박사학위논문
Ph.D. Dissertation

맥락 단위를 활용한 절차적 비디오 기반 작업 학습 향상

Enhancing Human Task Learning from Procedural Videos through Contextual Units

2026

양세린 (梁世隣 Yang, Saelyne)

한국과학기술원

Korea Advanced Institute of Science and Technology

박사학위논문

맥락 단위를 활용한 절차적 비디오 기반 작업 학습 향상

2026

양세린

한국과학기술원

전산학부

맥락 단위를 활용한 절차적 비디오 기반 작업 학습 향상

양세린

위 논문은 한국과학기술원 박사학위논문으로
학위논문 심사위원회의 심사를 통과하였음

2025년 12월 1일

심사위원장	김주호	(인)
심사위원	이탁연	(인)
심사위원	정준선	(인)
심사위원	송예일	(인)
심사위원	Amy Pavel	(인)

Enhancing Human Task Learning from Procedural Videos through Contextual Units

Saelyne Yang

Advisor: Juho Kim

A dissertation submitted to the faculty of
Korea Advanced Institute of Science and Technology in
partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Computing

Daejeon, Korea
December 1, 2025

Approved by

Juho Kim
Professor of Computer Science

The study was conducted in accordance with Code of Research Ethics[1].

[1] Declaration of Ethical Conduct in Research: I, as a graduate student of Korea Advanced Institute of Science and Technology, hereby declare that I have not committed any act that may damage the credibility of my research. This includes, but is not limited to, falsification, thesis written by someone else, distortion of research findings, and plagiarism. I confirm that my thesis contains honest conclusions based on my own careful research under the guidance of my advisor.

DCS

양세린. 맥락 단위를 활용한 절차적 비디오 기반 작업 학습 향상. 전산학부 . 2026년. 112+viii 쪽. 지도교수: 김주호. (영문 논문)
Saelyne Yang. Enhancing Human Task Learning from Procedural Videos through Contextual Units. School of Computing . 2026. 112+viii pages. Advisor: Juho Kim. (Text in English)

초 록

사람들은 절차적 비디오를 활용해 물리적 기술 습득부터 소프트웨어 조작에 이르기까지 다양한 복잡한 작업을 배우고 수행한다. 그러나 비디오는 절차의 전 흐름을 담고 있음에도, 선형적 형식 때문에 효율적인 학습에 필요한 구조가 부족하다. 중요한 정보들은 흐름 곳곳에 흩어져 있거나 묻혀 있어, 학습자가 필요한 구간을 찾거나 내용을 효율적으로 소화하기 어렵다. 비선형적이고 계속 변하는 워크플로를 가진 학습자들은 결국 핵심 정보를 추출하고, 맥락 속에서 해석하며, 이를 자신의 작업 환경에 대응시키는 데 상당한 노력을 들여야 한다. 이를 해결하기 위해 본 연구에서는 절차적 비디오에 절차의 무엇을, 어떻게, 왜 수행하는지를 정의하는 의미적 구조인 맥락 단위를 결합한다. 본 연구는 학습 과정을 이해 단계와 적용 단계로 나누고, 각 단계에 맞는 맥락 단위를 제안한다. 이해 단계에서는 학습자의 이해를 돕기 위해 영상 지식을 구조화하는 프레임워크를 제시하며, 흩어진 튜토리얼을 통합해 전체적 개요를 제공하거나 비선형 탐색을 지원하는 정보 유형 분류 체계 등을 소개한다. 적용 단계에서는 사용자 상호작용 데이터와 사용자 행동 및 의도 모델링이 소프트웨어 작업 중 실시간 피드백과 상황에 적절한 도움을 제공할 수 있음을 보여준다. 본 연구는 절차적 비디오가 세밀한 맥락 단위로 보강될 때, 초기 정보 탐색 단계부터 복잡한 작업 수행 단계까지 전 과정을 효과적으로 지원할 수 있음을 보여준다.

핵 심 낱 말 절차적 비디오, 태스크 학습, 상황 인지 시스템, 인간-인공지능 협업, 지능형 사용자 인터페이스

Abstract

People rely on procedural videos to learn and carry out complex tasks, from developing physical skills to operating software. Although videos capture the full flow of a procedure, their linear format lacks the structure needed for effective learning. Important details are scattered or buried within the stream, making it difficult to locate relevant segments or process information efficiently. Learners, whose workflows are non-linear and constantly shifting, must therefore invest substantial effort in extracting key details, interpreting them in context, and mapping them onto their own environments. To address this, I develop systems that augment procedural videos with contextual units—semantic structures that define the what, how, and why of a procedure. I propose contextual units aligned with each phase of the task-learning cycle across both the understanding and applying stages. In the understanding stage, I present frameworks for structuring video knowledge to enhance comprehension, including systems that consolidate scattered tutorials into holistic overviews and taxonomies that support non-linear navigation. In the applying stage, I show how user interaction data, along with user behavior and intent modeling, can provide real-time feedback and context-aware assistance during software tasks. My work shows that when procedural videos are augmented with granular contextual units, they can effectively support users from the initial stage of information gathering to the final stage of complex task execution.

Keywords Procedural Videos, Task Learning, Context-Aware Systems, Human-AI Collaboration, Intelligent User Interfaces

Contents

Contents i

List of Tables v

List of Figures vi

Chapter 1. Introduction 1

1.1 Procedural Videos 1

1.2 Contextual Units in Videos 1

1.3 Task Learning Cycle 2

1.3.1 Understanding: Knowledge Acquisition and Synthesis . 2

1.3.2 Applying: Knowledge Transfer and Execution 3

1.4 Contextual Units for Each Learning Phase 3

1.4.1 Understanding: Structuring Video Knowledge 3

1.4.2 Applying: Supporting Context-Aware Assistance 4

1.4.3 Summary 5

1.5 Contributions 6

1.6 Thesis Overview 7

Chapter 2. Related Work 8

2.1 Learning from Procedural Videos 8

2.1.1 Video Navigation and Skimming Techniques 8

2.1.2 Learning from Multiple Videos and Workflows 9

2.2 Structuring Instructional Video 9

2.2.1 Semantic Units and Information Types 9

2.2.2 Interaction Data for Understanding Instructional Content 10

2.3 Intelligent User Assistance 10

2.3.1 Modeling User Behavior from Software Usage Logs . . . 10

2.3.2 Context-Aware Assistance for Following Tutorials 10

2.3.3 Collaborative and Proactive AI Agents 11

2.4 Computer Vision Approaches to Video Understanding 11

2.4.1 Action Understanding in Procedural Videos 11

2.4.2 GUI and Software Workflow Video Understanding . . . 12

2.4.3 Video Question Answering 12

Chapter 3. VideoMix: Aggregating How-To Videos for Task-Oriented Learning 13
3.1 Motivation and Contributions 13
3.2 Formative Study 14
3.2.1 Method 14
3.2.2 Findings 15
3.2.3 Design Goals 17
3.3 VideoMix 17
3.3.1 System Interface 17
3.3.2 Technical Pipeline 18
3.3.3 Implementation 22
3.4 Technical Evaluation 22
3.4.1 Task Selection 22
3.4.2 Method 23
3.4.3 Results 24
3.5 User Study 24
3.5.1 Participants and Apparatus 24
3.5.2 Study Procedure 25
3.5.3 Results 25
3.6 Discussion and Future Work 28
3.7 Conclusion 30

Chapter 4. Beyond Instructions: A Taxonomy of Information Types in How-to Videos 31
4.1 Motivation and Contributions 31
4.2 Taxonomy of Information Types in How-to Videos 32
4.2.1 Methods 33
4.2.2 Taxonomy 34
4.3 Dataset 36
4.3.1 Method 37
4.3.2 Dataset: HTM-Type 37
4.4 Analysis 37
4.4.1 Method 37
4.4.2 Results 38
4.5 Taxonomy as Analytical Framework 40
4.6 Exploratory User Study 41
4.6.1 Research Probe 42
4.6.2 Study Procedure 42

4.7 Results 44
4.7.1 How Taxonomy Was Used in Each Task 44
4.7.2 Effect of Taxonomy on Video-watching Experience . . . 46
4.7.3 Perception Toward Taxonomy 46
4.8 Discussion 46
4.9 Conclusion 51

Chapter 5. SoftVideo: Improving the Learning Experience of Software Tutorial Videos with Collective Interaction Data 52

5.1 Motivation and Contributions 52
5.2 Data collection study 53
5.2.1 System for Data Collection 54
5.2.2 Participants 54
5.2.3 Task 54
5.2.4 Procedure 54
5.2.5 Results 55
5.3 Data Analysis Pipeline 56
5.3.1 Measures 56
5.3.2 Methodology 58
5.3.3 Results 61
5.4 SoftVideo 61
5.4.1 Step Information 62
5.4.2 Real-time Feedback 63
5.4.3 Help Suggestions 63
5.4.4 Implementation 64
5.5 User Evaluation 64
5.5.1 Participants 64
5.5.2 Study Procedure 65
5.6 Results 65
5.6.1 Step Information 65
5.6.2 Real-time Feedback 66
5.6.3 Help Suggestions 67
5.6.4 Other Feedback 68
5.7 Discussion, Limitations, and Future Work 68
5.8 Conclusion 70

Chapter 6. GUIDE: A Benchmark for Understanding and Assisting Users in Open-Ended GUI Tasks 71
6.1 Motivation and Contributions 71
6.2 GUIDE Benchmark 73
6.2.1 Video Collection 73
6.2.2 Benchmark Tasks 74
6.3 Experiments 79
6.3.1 Experimental Setup 79
6.3.2 Evaluation Tasks 79
6.3.3 Results 81
6.4 Conclusion 85

Chapter 7. Discussion 86
7.1 The Dynamic Nature of Contextual Units 86
7.2 Design Principles for Selecting Effective Contextual Units . . . 87
7.3 User Modeling for Adaptive Procedural Support 88
7.4 Generalizability of Contextual Units 88

Chapter 8. Conclusion and Future Work 90
8.1 Summary of Contributions 90
8.2 Future Directions 90
8.2.1 Personalized and Adaptive Videos 90
8.2.2 Generative Instructional Media 90
8.2.3 Collaborative AI Agents for Task Learning 91
8.2.4 Longitudinal Support for Skill Development 91

Bibliography 92

Acknowledgments 109

Curriculum Vitae 111

List of Tables

3.1 Results of the technical evaluation of our DAI module. Our pipeline maintained step accuracy across Relevancy, Logical Flow, and Completeness (with no statistically significant differences), while capturing a significantly more diverse range of possible approaches, both (1) when considering only the approaches and (2) across all outcome types (*: p¡0.05, **: p¡0.01). 23

4.1 Definition and examples of information types in our taxonomy. Minor errors from Speech-to-Text results in example sentences are corrected. 35

4.2 Example systems that support video navigation and information types associated with each system. 41

4.3 Possible applications of the taxonomy in video authoring, viewing, and analysis. 48

5.1 Tutorial videos used in the data collection study. 53

5.2 The number of collected logs for each tutorial. 56

5.3 Average time taken to complete each tutorial. 56

5.4 Definition of six measures that portray the difficulty of each step. 56

5.5 Definition of three measures related to relevancy of each step. 57

5.6 Mean values of the six difficulty-related measures among all steps. In general, novice users show more behavior of having difficulties than experienced users. For each measure, the table shows if the difference between the novice and experienced groups was statistically significant (*: p¡.05, **: p¡.01, Mann-Whitney Test) for each measure. 61

5.7 Icons that depict the difficulty of each step, and their corresponding meanings and measures. 63

6.1 Comparison of GUIDE with existing GUI video understanding datasets. GUIDE differs from existing benchmarks by (*i*) collecting screen recordings from novice users, (*ii*) capturing how they naturally behave in open-ended tasks with a focus on behavior understanding, and (*iii*) evaluating systems based on human user needs rather than task automation. . . 72

6.2 Example instances for the (1) User Behavior State Detection task. 77

6.3 Example instances for the (2) Intent Prediction task. 78

6.4 Example instances for the (3) Help Prediction task. For the Help Need Detection task, the top three instances are labeled as *help needed*. 80

6.5 Evaluation results on accuracy across (1) Behavior State Detection, (2) Intent Prediction, and (3) Help Prediction. 82

6.6 Results for **Help Need Detection** on accuracy, precision, recall, and F1-score across three conditions (default, with behavior state, with behavior state and intent). 84

6.7 Evaluation of **Intent Prediction** and **Help Content Prediction**, with Accuracy (Acc) and Multi-Binary Accuracy (MBAcc). 85

List of Figures

1.1 Overview of the Task Learning Cycle. The process is divided into two primary stages: **(1) Understanding**, which comprises the **Exploration** and **Comprehension** phases for gathering and processing information, and **(2) Applying**, which involves the **Following** and **Autonomous** phases for transferring knowledge to the target environment. 2

1.2 VideoMix organizes multiple how-to videos into a structured hierarchy. It first clusters videos by Outcome, then determines diverse Approaches for each result. Finally, it aligns video segments to specific Steps, extracting distinct Methods, tips, and notes. 4

1.3 Taxonomy of information types in how-to videos. 4

1.4 (a) The SoftVideo timeline visualizes step-wise difficulty using data-driven icons. (b) Users can inspect these icons to understand specific challenges. (c) The system provides real-time feedback on execution progress, alerting users if a step is skipped. 5

1.5 An example of the GUIDE benchmark, which jointly models three tasks: Behavior State Detection, Intent Prediction, and Help Prediction, to interpret what the user is doing, aiming to achieve, and whether and what they may need assistance with during open-ended software tasks. 5

3.1 VideoMix interface on the Overview page for the task “Build a Desk”. (A) Users begin by selecting the task they want to learn. (B) VideoMix then presents video search results categorized by outcome types. (C) For each outcome type, users can choose from standard, simple, or complex approaches. (D) Based on the chosen approach, VideoMix displays the necessary requirements, such as materials, ingredients, and tools. Finally, (E) users can see a list of steps and a brief description of each step that makes up the chosen approach.[1] 19

3.2 VideoMix interface on the Details page for the task “Build a Desk”. (A) The interface displays the list of steps for the chosen approach. (B) For each step, users can explore different methods, such as tools or techniques, to complete the step. (C) When a method is selected, VideoMix presents video snippets relevant to that method. (D) Users can easily switch between different videos for the selected method, with the corresponding time frame playing automatically. (E) Additionally, users can view tips and notes extracted from the videos.[2] 20

3.3 Illustration of our Dynamic Approach Identification (DAI) module, which captures a variety of approaches to accomplish a task. (a) The process begins by extracting step information from the first video using GPT-4o. This initial step taxonomy is then applied to the next video, where additional steps are identified, refining the taxonomy. This iterative process continues for all videos, progressively refining the step taxonomy with each comparison. (b) Once the final step taxonomy is established, it is reapplied to each video to detect relevant steps and align segments accordingly. Note that not all steps may be present in each video. (c) After extracting step information from each video using the common taxonomy, the system identifies standard, simple, and complex approaches based on the number of videos that follow each approach and the number of steps within each approach. 21

3.4 Participants felt they were more successful and efficient with VideoMix, and found VideoMix to be more useful when learning about the task compared to the baseline. There were no statistically significant differences in mental demand, effort, and frustration (*: p¡0.05). . 26

3.5 Participants' ratings on the usefulness of each information piece in understanding the task. Overall, they found the information provided by VideoMix—including outcome types, requirements, different approaches, step details, methods, and tips and notes—to be helpful in gaining a better understanding of the task. 27

4.1 Distribution of Categories and Types of all videos in HTM-Type. Categories are denoted above the types using group brackets. Only proportions greater than 1.5% are written in text. `Instruction` makes up 39.8% of the total video, suggesting that the majority of the video contains information that does not directly give actions for the user to follow. The results illustrate the large diversity of information types in how-to videos. 38

4.2 The number of labels for the category along normalized time. *Greeting*, *Overview*, *Conclusion*, and *Miscellaneous* show clear positional preferences while *Method*, *Supplementary*, *Explanation* and *Description* are widely distributed. 40

4.3 Our research probe used in the user study. (a) Users can see the video. (b) Each sentence of the script is shown with its timestamp and information type. Each type label is color-coded based on the category. (c) The same information is shown in the timeline. (d) When users hover over each segment, they can see the type and (e) its definition in the Filter panel. Users can filter segments based on their type or category in the Filter panel. Only the filtered segments are shown in the transcript panel and the timeline. 43

4.4 Helpfulness (left) and Importance score (right) of each category in the Summarize and Follow task. 44

5.1 An example session of the data collection study. A participant is following the tutorial video (on the left) on their software (on the right). 55

5.2 An example scenario where the relevant step of step 15 is step 12. After a user followed the step 12, 13, and 14, he is now on step 15. However, the user was not able to complete it. The user jumped back to step 12 and then followed it again. Then, he came back to step 15 and followed the step. (red: followed, gray: watched but not followed, blue: followed again). 58

5.3 Overview of SoftVideo. Along with the software tutorial video, SoftVideo provides (a) a timeline where users can see the action name, its length, and the estimated difficulty. (b) Users can receive real-time feedback on their progress. If a user followed a step, the circle will be filled. (c) SoftVideo detects users' confusing moments. Once detected, it provides users with suggestions such as (d) slowing down the pace, (e) replaying the step, or (f) seeing relevant steps. Users see customized information based on (g) the expertise level they enter. 62

5.4 (Left) A user is following the tutorial video. Once the system detects that the user may be confused or struggling, (Right) SoftVideo presents action suggestions as help. Users can also proactively (a) request to see the help (b) or close the help. 64

6.1 Overview of the three core tasks in the GUIDE benchmark. (1) **User Behavior State Detection** identifies the user's current behavioral mode (e.g., *Exploration and Decision-Making*). (2) **Intent Prediction** infers what the user is trying to achieve (e.g., *Create a progress bar*). (3) **Help Prediction** determines whether the user needs assistance and, if so, what kind of help is relevant (e.g., *Get a guide on how to use text effects*). Together, these tasks enable a comprehensive understanding of user behavior and assistance needs in software GUI environments. We evaluate MLLMs on their ability to infer these solely from the visual input, without access to the demonstrator's narration — a setting that closely reflects real-world use. 73

6.2 Statistics of the GUIDE dataset. 74

6.3 Distribution of screen recording video lengths. 74

6.4 Our proposed taxonomy of user behavior states in GUI-based software tasks, organized into four main phases: **Planning**, **Execution**, **Problem-Solving**, and **Evaluation**. Each phase captures distinct patterns of user cognition and interaction, from initial goal formulation to iterative action, troubleshooting, and reflection. 75

6.5 Normalized confusion matrix for user behavior state classification. The most common errors occur when *Frustration* or *Debugging* is misclassified as *Performing Actions* or *Exploration and Decision-Making*. 83

6.6 Accuracy trends for Gemini-2.5-Flash [62] and InternVideo2.5-8B [172] across the tasks in the online setting, where models are given progressively more of the video segment (25%, 50%, 75%, and 100%). Both models show steady performance gains as they see more segments, while Gemini-2.5-Flash shows larger and more consistent gains. 85

7.1 The same user query can be supported differently depending on the learning phase, with systems adapting contextual units to the user's current goal. 87

Chapter 1. Introduction

1.1 Procedural Videos

Procedural videos have become a dominant medium for knowledge acquisition. From mastering cooking skills to operating complex software, millions of learners turn to platforms like YouTube to acquire new skills. The appeal of video lies in its ability to capture the dynamic flow and rich visual details—showing not just what to do, but how to do it, including subtle visual cues, timing, and technique that static text often fails to convey.

However, despite the richness of this medium, the format of video remains inherently linear and unstructured. Unlike text, which is indexed and organized into paragraphs and headers, video is a continuous temporal stream, which allows access to only one point at a time. Critical information is often fragmented across the video stream, or buried within rapid and visually dense demonstrations.

In contrast, the learner's workflow is inherently non-linear. Rather than consuming information sequentially, users engage in dynamic behaviors—jumping, repeating, and pausing—as their focus shifts from broad exploration to targeted troubleshooting. This discrepancy is most pronounced when users attempt to transfer video instructions to their own environments, a task that requires an iterative, back-and-forth process.

Consequently, learners face substantial difficulties. They struggle to skim content for relevant segments, to synthesize information across scattered tutorials, and to adapt demonstrations to their own constraints or goals. As a result, much of the cognitive burden falls on learners as they extract relevant details, interpret them within context, and translate them into actions.

1.2 Contextual Units in Videos

To bridge the gap between the linear nature of video and the dynamic needs of the learner, this thesis introduces the concept of **Contextual Units** — semantic structures that capture the distinct aspects of a procedure, including the "what," "how," and "why". Traditionally, approaches have attempted to mitigate the continuous nature of video through temporal segmentation, where the "step" serves as the fundamental unit of organization [56, 33]. This method divides the timeline into chronological chapters, enabling users to navigate by step. Other systems have expanded this by utilizing "objects" (e.g., tools or ingredients in a recipe video) as primary units of interaction [24].

While these traditional units effectively define the "what" of a procedure and support basic navigation, they remain limited in scope. They are primarily designed for passive watching or sequential following, failing to accommodate the dynamic, evolving workflow of learners. The definition of a meaningful unit must evolve depending on the learner's current stage. Consequently, supporting users beyond simple navigation requires identifying new types of contextual units that address the complex needs arising across different stages of task learning.

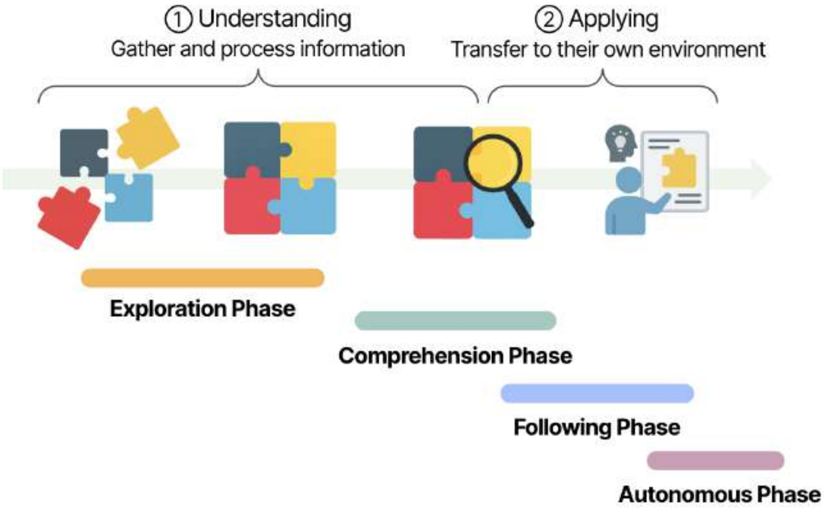


Figure 1.1: Overview of the Task Learning Cycle. The process is divided into two primary stages: **(1) Understanding**, which comprises the **Exploration** and **Comprehension** phases for gathering and processing information, and **(2) Applying**, which involves the **Following** and **Autonomous** phases for transferring knowledge to the target environment.

1.3 Task Learning Cycle

To introduce appropriate contextual units for each stage of task learning, we begin by outlining the Task Learning Cycle. Learning from procedural videos is not a linear act of consuming content but a dynamic, iterative process that evolves as learners move from initial curiosity to proficient execution. This dissertation conceptualizes this process as the **Task Learning Cycle**, which consists of two primary stages: *Understanding*, where knowledge is acquired and synthesized, and *Applying*, where that knowledge is transferred into active practice. Each stage is further composed of distinct sub-phases, each characterized by unique user behaviors and information needs.

1.3.1 Understanding: Knowledge Acquisition and Synthesis

The first stage focuses on the learner's interaction with the video content itself. Before a task can be performed, the learner must navigate a vast landscape of available resources to construct a mental model of the procedure.

- **Exploration Phase**: The cycle begins with a broad search. Learners often start with a high-level goal (e.g., "how to build a desk" or "how to edit a portrait image"), but are confronted with hundreds of potential tutorials. In this phase, users are not yet committed to a single instructional path. Instead, they scan multiple videos to survey the landscape of possibilities, comparing different *outcomes* (e.g., rustic vs. modern designs) and assessing various *approaches* (e.g., beginner tools vs. professional equipment). The primary challenge here is synthesizing scattered information to select the workflow that best matches their constraints.
- **Comprehension Phase**: Once a specific resource is selected, the learner shifts to deep processing. The goal transitions from filtering content to internalizing instructions. However, comprehending a procedural video involves more than passive watching; users actively seek specific types of

information—such as the rationale behind a step, the tools required, or warnings about potential pitfalls. They must deconstruct the continuous video stream to locate these granular details, often navigating non-linearly to re-watch complex segments or skip known information.

1.3.2 Applying: Knowledge Transfer and Execution

The second stage marks the shift from the video player to the user's work environment. Here, the video serves as a reference utility while the user engages in the physical or digital execution of the task.

- **Following Phase**: In this phase, learners attempt to replicate the demonstrated steps in their own environment (e.g., software application). This process is mentally demanding, as users must constantly switch contexts between the instructional video and their workspace. The challenge lies in synchronization: users struggle to match the video's pacing with their own speed, frequently pausing to interpret actions or rewinding to recover from missed details. They must map the demonstration to their specific context, often dealing with differences in software versions or available assets.

- **Autonomous Phase**: As users gain proficiency or encounter open-ended problems, the guidance of a tutorial becomes insufficient. They move beyond simple replication into independent exploration, where they inevitably face errors or uncertainties. Systems must recognize the user's evolving intent (e.g., "why isn't this layer blending?") and behavioral state (e.g., frustration or confusion) to provide context-aware support that complements their autonomous workflow.

1.4 Contextual Units for Each Learning Phase

1.4.1 Understanding: Structuring Video Knowledge

The first thread of my thesis focuses on the Understanding phase, where the primary challenge is the lack of structure in raw video data. Learners often need to synthesize information from multiple sources or navigate non-linearly within a single video to grasp the "big picture" of a task. To support this, I propose frameworks that organize video content into meaningful informational units.

(1) Exploration: Aggregating Multiple How-To Videos for Task-Oriented Learning

When learners start a new task, they often need to watch multiple videos to understand the landscape of possible outcomes and methods. Navigating these scattered resources is time-consuming. VideoMix [190] is a system that aggregates information from multiple how-to videos to provide a holistic understanding of a task. By organizing content into contextual units of Outcomes, Approaches, and Methods, VideoMix allows users to compare different workflows (e.g., "Standard" vs. "Simple") and methods to achieve a step, enabling them to form a mental model of the task before committing to a specific tutorial.

(2) Comprehension: A Taxonomy of Information Types in How-to Videos

Once a learner selects a video, they face a vast amount of information—not just instructions, but also rationale, tips, and warnings. I present a comprehensive taxonomy of information types in how-to videos [188]. Through an analysis of 120 videos, we identified 21 distinct informational units (e.g., Justification, Status, Tip). We demonstrate that exposing these units allows learners to navigate directly

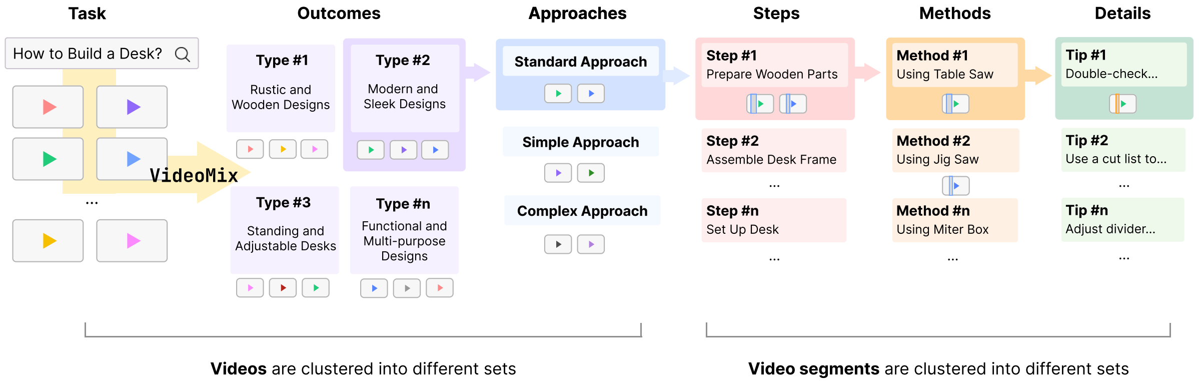
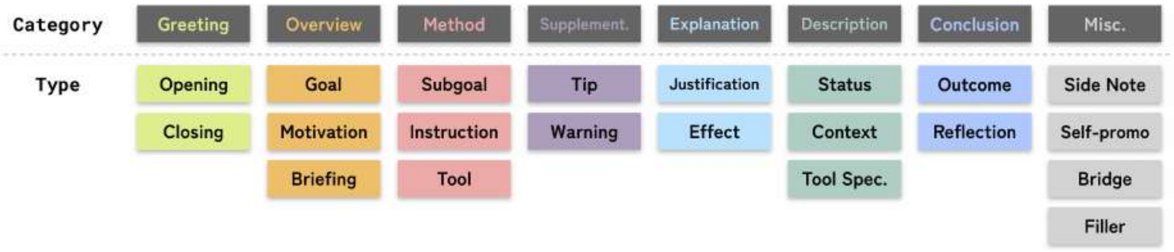


Figure 1.2: VideoMix organizes multiple how-to videos into a structured hierarchy. It first clusters videos by Outcome, then determines diverse Approaches for each result. Finally, it aligns video segments to specific Steps, extracting distinct Methods, tips, and notes.



Category	Greeting	Overview	Method	Supplement.	Explanation	Description	Conclusion	Misc.
Type	Opening	Goal	Subgoal	Tip	Justification	Status	Outcome	Side Note
	Closing	Motivation	Instruction	Warning	Effect	Context	Reflection	Self-promo
		Briefing	Tool			Tool Spec.		Bridge
								Filler

Figure 1.3: Taxonomy of information types in how-to videos.

to the knowledge they need, supporting non-linear consumption that goes beyond simple step-by-step playback.

1.4.2 Applying: Supporting Context-Aware Assistance

The second thread focuses on the Applying phase, where the challenge shifts to the friction between watching a video and executing the task in the user's environment. Here, I show how user interaction logs and user demonstration videos can reveal meaningful contextual units that enable real-time feedback and context-aware assistance.

(3) Following: Improving the Learning Experience of Software Tutorial Videos with Collective Interaction Data

When users transfer instructions from a video to their own software, such as Photoshop, they often struggle to map the demonstration to their specific context. SoftVideo [191] is a system that improves the learning experience by utilizing collective interaction data. By analyzing how previous learners interacted with both the video and the software, the system computes contextual units of Difficulty and Relevancy for each step. This allows the system to provide real-time feedback, such as detecting moments of struggle, warning users about commonly missed steps, or suggesting relevant video segments when they become stuck.

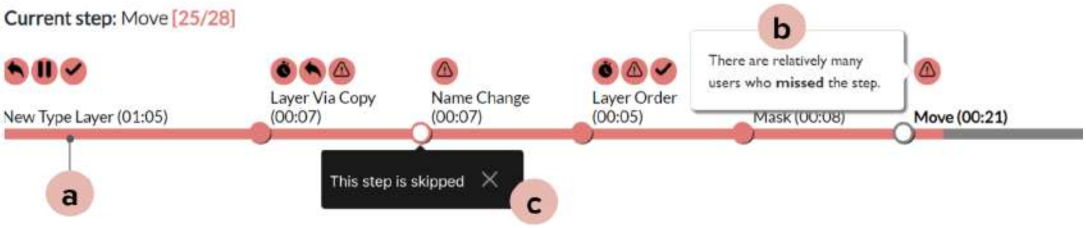


Figure 1.4: (a) The SoftVideo timeline visualizes step-wise difficulty using data-driven icons. (b) Users can inspect these icons to understand specific challenges. (c) The system provides real-time feedback on execution progress, alerting users if a step is skipped.

(4) Autonomous: Understanding and Assisting Users in Open-Ended GUI Tasks

Finally, users move beyond following instructions to engaging in open-ended, self-directed tasks. In this setting, the user's own demonstration video becomes the target video, and an intelligent agent must "watch" the continuous stream of screen activity to infer the user's needs. GUIDE introduces a benchmark for evaluating multimodal AI systems on their ability to perceive high-level contextual units from this visual stream, such as user behavior state and intent. For AI agents to serve as truly effective collaborators, they must be able to infer these human-centric units and provide context-aware assistance.

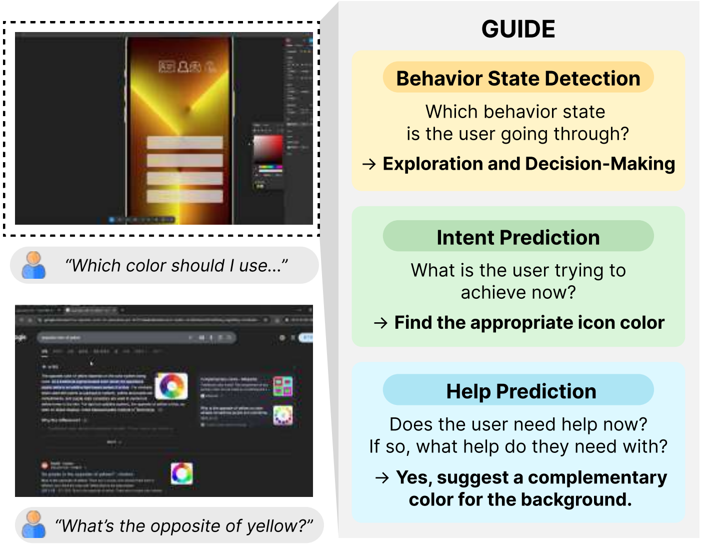

"Which color should I use..."
"What's the opposite of yellow?"
GUIDE
Behavior State Detection
Which behavior state is the user going through?
→ Exploration and Decision-Making
Intent Prediction
What is the user trying to achieve now?
→ Find the appropriate icon color
Help Prediction
Does the user need help now? If so, what help do they need with?
→ Yes, suggest a complementary color for the background.

Figure 1.5: An example of the GUIDE benchmark, which jointly models three tasks: Behavior State Detection, Intent Prediction, and Help Prediction, to interpret what the user is doing, aiming to achieve, and whether and what they may need assistance with during open-ended software tasks.

1.4.3 Summary

In summary, effectively supporting human task learning from procedural videos requires more than generic video segmentation. As learners' goals shift—from high-level scanning in the **Exploration** phase to precise troubleshooting in the **Autonomous** phase—the contextual units needed to structure and interpret the video must evolve accordingly. This dissertation demonstrates that by identifying the specific **Contextual Unit** tailored to each learning phase, we can transform the linear video stream into an adaptive interface that scaffolds the user's journey from initial understanding to proficient execution. The specific contextual units identified for each phase are summarized as follows:

1. **Understanding**: The initial stage of knowledge acquisition where learners gather and synthesize information.
    - **Exploration Phase**: Users scan multiple videos to survey the landscape of a task. Here, the relevant units are macro-level structures like *Outcomes*, *Approaches*, and *Methods*(Chapter 3).
    - **Comprehension Phase**: Users dive deep into a single video. Here, the relevant units are granular *Information Types*, such as justifications and tips (Chapter 4).
2. **Applying**: The active stage of task execution where learners transfer knowledge to their own environment.
    - **Following Phase**: Users perform the task in their software. Here, the relevant units are data-driven metrics of *Difficulty* and *Step Relevancy* (Chapter 5).
    - **Autonomous Phase**: Users work independently in their own environment. In this stage, user demonstration videos require higher-level contextual units, such as *Behavior States* and *Intent*, to support effective assistance (Chapter 6).

1.5 Contributions

This thesis makes two primary technical contributions:

1. Frameworks and pipelines for decomposing unstructured, linear video streams into semantic contextual units to enhance learner comprehension and navigation.
2. Data-driven methods and evaluation benchmarks that leverage these units to scaffold active task execution, providing real-time feedback and enabling context-aware human-AI collaboration.

These contributions are instantiated through a series of systems, taxonomies, and benchmarks designed to support the full Task Learning Cycle. The structural frameworks (VideoMix, Beyond Instructions) aggregate and categorize scattered video content to support the *Understanding* phase, while the data-driven systems and benchmarks (SoftVideo, GUIDE) utilize collective interaction data and user intent modeling to support the *Applying* phase. Together, these approaches demonstrate that moving beyond the raw video stream to a structured representation of contextual units allows for intelligent systems that adapt to the learner's evolving needs—from initial exploration to complex problem-solving.

The contributions are enabled by uniquely combining and extending the following methodological foundations: **Human-Computer Interaction**, which informs the design of user-centered methods that support the comprehension and application of procedural knowledge; and **Video Understanding and AI**, which enable the automatic extraction of semantic structures and the inference of user states from visual data.

Thesis statement: Augmenting procedural videos with granular contextual units can effectively support the full lifecycle of human task learning.

1.6 Thesis Overview

- **Chapter 2** reviews prior work across four areas foundational to this thesis: (1) learning from procedural videos, (2) structuring instructional video content, (3) intelligent user assistance, and (4) computer vision approaches to video understanding.
- **Chapter 3** presents *VideoMix*, a system that supports the exploration phase of learning by aggregating multiple how-to videos into a holistic overview. It introduces a pipeline to extract contextual units of outcomes, approaches, and methods, allowing users to compare workflows before diving into specific instructions.
- **Chapter 4** introduces *Beyond Instructions*, a comprehensive taxonomy that structures the information types in how-to videos. By analyzing 120 videos, this work identifies 21 granular information types (such as justifications and tips) that allow systems to support non-linear video navigation beyond step-by-step playback.
- **Chapter 5** describes *SoftVideo*, a system designed to scaffold the execution phase where users transfer video knowledge to their own software environment. It demonstrates how collective interaction data can serve as a contextual unit for estimating step difficulty and providing real-time, context-aware feedback.
- **Chapter 6** presents *GUIDE*, a benchmark for the assistance phase that evaluates multimodal AI models on their ability to collaborate with users. It introduces high-level user context—specifically behavior states and intent—as essential units for shifting AI agents from blind automation to user-aware collaboration.
- **Chapter 7** discusses the broader implications of transforming linear video streams into structured contextual units, synthesizing findings across the four systems to propose design guidelines for future intelligent learning systems.

Chapter 2. Related Work

This thesis builds upon four key areas of prior research: **(1) learning from procedural videos**, which examines how learners navigate and extract meaningful information from videos; **(2) structuring instructional content**, which investigates semantic units and interaction signals that reveal how procedural knowledge is organized within videos; **(3) intelligent user assistance**, which explores systems that adapt to users' context and behavior to provide timely support during task execution; and **(4) computer vision approaches to video understanding**, which enable the modeling of actions and workflows in procedural videos.

2.1 Learning from Procedural Videos

2.1.1 Video Navigation and Skimming Techniques

How-to videos provide rich explanations of how to complete a task. However, the linear nature of the video makes it difficult for users to navigate or skim through the content [141, 54, 34]. For example, it is hard to locate a specific point of interest in videos without navigating over a time scale. Researchers have proposed several approaches to overcome such limitations. One of the popular approaches is to segment a video into meaningful sections [192, 167, 175, 56, 84, 149, 127, 141, 34, 56, 84, 149]. It helps users navigate the video based on semantics and locate a section of interest. Truong et al. have introduced two-level hierarchical makeup videos, where they organize a set of actions into spatial locations [167]. Similarly, VideoWhiz organized steps in recipe videos by reflecting the dependencies between the steps [127].

Another approach is to identify conceptual objects introduced in videos, which allows users to navigate a video based on objects or concepts of interest [24, 107, 122]. Specifically, RubySlippers [24] focused on a setting where users' hands are occupied with physical activities, which it supports with keyword-based voice commands for navigating videos. A data-driven approach has been introduced as well to improve video navigation. Researchers found that interaction traces of other users help identify points of importance or confusion [82]. Finally, transcript-based navigation approaches have allowed users to efficiently search the content [139, 82], give feedback on videos [140], or edit videos [45, 70, 166, 16].

To better convey this information within the video interface, several systems present it in a mixed-media format, displaying screenshots alongside corresponding descriptions such as step labels [141, 139, 34, 167]. This presentation format helps users digest the content more efficiently, making it easier to skim and navigate through the material.

In summary, existing methods for video navigation are based on the script, conceptual objects, section, or interaction traces. While the script and conceptual objects allow users to navigate in a finer-grained way, it lacks in supporting navigation in a holistic view. On the other hand, while section and interaction traces allow users to see the overall flow of videos, it does not support detailed navigation. Beyond Instructions [188] presents a novel unit for video navigation, information types, which allows users to see the overall composition of videos as well as navigate at a shorter segment level. It shows how information types enable efficient navigation through a research probe.

2.1.2 Learning from Multiple Videos and Workflows

Learning from multiple resources can foster a deeper understanding of a subject [116, 106]. FollowUs [92] demonstrated the effectiveness of offering multiple demonstrations of a software tutorial performed by different users, providing various insights and allowing learners to pick up on pieces from different tutorials. To facilitate this, researchers have developed systems that enable the comparison of hundreds of cooking recipes [23] or software workflows [88], as well as computational pipelines that capture the diversity of these demonstrations [25, 170]. A similar approach has been explored in the context of multi-document analysis, where systems were proposed to effectively collect and organize information from multiple relevant documents [55, 64].

In video-specific research, several systems have been proposed to facilitate multi-video analysis. For example, Surch [81] enables structured search and comparison of surgical videos, while Video Lens [114] offers interactive search and exploration of baseball videos. Work in this space has explored techniques for comparing instructional steps across videos, including detecting differences between two demonstrations of the same step [124] and navigating to alternative videos that illustrate different ways of performing that step [8]. These approaches expand multi-video navigation by helping learners understand procedural variations across demonstrations.

When presenting information from multiple videos, it is important to organize the content in a structured manner to avoid overwhelming users. Prior work has explored improving the browsing of multiple video snippets by organizing frames along meaningful dimensions for video editing [101], or content exploration [114, 200]. However, these approaches typically focus on visual frames, sorting them in latent space, or rely on metadata for a specific application. In instructional how-to videos, however, verbal content also carries critical information [188, 189], as these videos often contain a richer depth of knowledge, delivered through both visual and verbal channels. Building on these ideas, VideoMix [190] enhances multi-video skimming of how-to videos, helping users process and synthesize complex, detailed information from multiple sources.

2.2 Structuring Instructional Video

2.2.1 Semantic Units and Information Types

Instructional videos contain rich semantic cues that help convey how a procedure unfolds, but these cues are often implicit and embedded within an unstructured visual stream. Prior work has examined various ways to surface these units to support learning. In how-to videos, researchers have identified meaningful components such as subgoals, tools, and intermediate outcomes to structure the procedural flow and support navigation [175, 167, 84, 127]. These units highlight what the user is trying to achieve and what resources are needed, providing coarse structure for understanding the task.

Beyond structural cues, several studies have explored the semantic content within instructional videos. For instance, analyses of narrated how-to videos have classified transcript sentences by their visual relevance to surface which parts of narration are directly grounded in the demonstration [65, 119]. Other systems have leveraged scene- or concept-level markers to support video authoring and editing, enabling users to annotate or organize content based on the nature of each segment [35]. These efforts show that identifying semantic units can support multiple tasks such as segmentation, editing, and browsing.

While these approaches advance the understanding of instructional content, they focus primarily on isolated unit types such as subgoals, scene markers, or visual anchors. Beyond Instructions [188] investigates the broader landscape of information types present in how-to videos, providing a more comprehensive view of the semantic elements that shape procedural understanding.

2.2.2 Interaction Data for Understanding Instructional Content

A stream of research has analyzed interaction data of educational videos to gain insights into learners' understanding of the video. A number of work analyzed interaction sequences to relate with learners' engagement and performance [18, 157, 97, 86, 89, 18, 63]. Another stream of research has analyzed video interaction data to reveal meaningful insights of the videos such as perceived difficulty [98] or important moments of the video [82, 39]. Kim et al. [83] have analyzed dropouts and peaks of interactions in different types of videos and suggested design implications for better video learning experiences. Li et al. [97] have analyzed in-video interactions together with a survey about perceived video difficulty to find relevant video interactions that indicate a student has experienced difficulty. However, it is still challenging to fully estimate a users' state with only video interaction data, especially in procedural tasks. Even if a user watches an entire tutorial, it remains unclear whether they were actually able to follow the steps or complete the task successfully. SoftVideo [191] addresses this limitation by analyzing synchronized interaction data from both the tutorial video and the target software, enabling the identification of meaningful signals such as in-step difficulties and relevant step relationships.

2.3 Intelligent User Assistance

2.3.1 Modeling User Behavior from Software Usage Logs

Software usage logs have been used to uncover patterns in how people perform tasks, providing a foundation for intelligent assistance. Prior work has analyzed application logs to identify frequent tasks or recommend workflows by comparing usage patterns across users [46, **?**, 125, 170]. Other research has classified sequences of commands to offer high-level overviews of user workflows and support semantic navigation through complex task histories [44, 110, 27].

Researchers have also incorporated usage-log analysis into user interfaces to surface helpful cues. For example, Patina [113] visualizes collective usage patterns of UI elements to help users work more efficiently. Such approaches demonstrate how behavioral traces can reveal user goals, task structures, and moments of difficulty—signals that are essential for building assistance systems that adapt to user needs. SoftVideo [191] extends this direction by analyzing synchronized video and software logs to identify step difficulty and relevancy, enabling more responsive and context-aware support during task execution.

2.3.2 Context-Aware Assistance for Following Tutorials

A line of work has explored how systems can assist users as they follow tutorial videos, particularly in software environments where learners often encounter context mismatches, such as interface differences between the video and their application [186, 145]. To mitigate these issues, systems like ReMap [57] and Replay [58] surface contextually relevant video segments based on the user's current software state, reducing navigation burden and helping users locate the most applicable instructions. Other approaches track a learner's progress across both the tutorial video and the target application, automatically adjusting playback or synchronizing the two contexts to support smoother task execution [148, 148, 129].

For physical how-to tasks, where users' hands are often occupied (e.g., using tools while watching video), researchers have investigated voice-based video control as an alternative interaction modality [26, 203, 102]. These systems highlight challenges in conversational video interaction, such as uncertainty from unseen content. RubySlippers [24] addresses some of these issues by enabling keyword-based navigation that lets learners quickly jump to relevant segments while staying engaged in their task.

Beyond video-specific assistance, related work has examined real-time suggestion mechanisms that guide users as they perform complex tasks. ViZig [180] and LectureScape [82] help learners locate important regions in educational videos by surfacing anchor points derived from collective interaction patterns. In broader interfaces, systems such as Adaptive Hypermedia [20], Ephemeral Adaptation [53], and Patina [113] personalize or adapt UI elements in response to user behavior. These methods collectively demonstrate how timely, context-aware assistance can reduce cognitive effort, anticipate user needs, and support learners as they navigate complex workflows.

2.3.3 Collaborative and Proactive AI Agents

Graphical User Interface (GUI) agents show strong potential for supporting users in complex workflows by automating tasks toward a given goal [61, 104, 201]. However, agents that fully automate interface operations can conflict with the needs of users in creative or analytical settings, where retaining control, exploring alternatives, and iterating on ideas are essential parts of the workflow. To address this, recent research has shifted toward assistive GUI agents that collaborate with users by understanding context and offering timely support. Several works have explored inferring user goals and intent in both web [142] and software environments [14, 60, 204] to better align assistance with user needs. For example, Zhao et al. [204] introduce ProactiveVA, a visual analytics agent that monitors user interactions and leverages LLMs to detect when users may be stuck, providing context-sensitive suggestions or guidance.

Recent works explore this shift toward collaboration and contextual support. CowPilot [71] proposes a mixed-initiative framework that enables users to share control with an autonomous web navigation agent, improving efficiency while preserving agency. In programming settings, proactive assistants like Codellaborator [151] and NeedHelp [28] demonstrate how real-time intervention can aid users when well-timed. Studies on software applications [78] show users prefer AI agents that guide them rather than take over entirely, reinforcing the need for transparency and shared control. ProMemAssist [152] further highlights the benefits of modeling user cognition (e.g., working memory) to deliver timely, non-intrusive support. These findings echo broader discussions on autonomy levels [51] and the importance of aligning agent behavior with human preferences [99, 79]. GUIDE builds on these insights, evaluating how well current multimodal models can perceive a user's state and intentions in GUI workflow recordings and decide if and how to assist, aiming to push GUI agents toward true user-aware collaboration.

2.4 Computer Vision Approaches to Video Understanding

2.4.1 Action Understanding in Procedural Videos

Instructional videos provide step-by-step guidance toward achieving task goals, containing hierarchical and procedural knowledge. To facilitate procedural video understanding, various datasets have been introduced [205, 171, 208, 161, 158, 196, 169, 120, 90]. These datasets are annotated with temporal segment boundaries and the actions performed within each segment, enabling a range of video understanding tasks such as video or moment retrieval [179, 120, 94, 12], video captioning [182, 3, 154, 103, 90, 168], and action recognition or localization [22, 15, 165, 91, 77, 138, 132, 30].

While these datasets have advanced video understanding, they primarily capture what actions occur, leaving open the challenge of modeling how those actions are performed. This distinction is central to procedural learning, where skill hinges on subtle variations in pace, technique, and control [160, 156, 178]. Emerging work in fine-grained action understanding begins to address this challenge by modeling verb–adverb relationships that differentiate manners of execution, such as "slice slowly" versus "slice quickly" [47, 121, 48]. These developments point toward richer representations that capture the nuance necessary for supporting procedural understanding.

2.4.2 GUI and Software Workflow Video Understanding

Several benchmarks evaluate video understanding in the context of GUI and software workflows. Early work by Li et al. [95] collected Photoshop tutorial videos to understand screencast videos. More recent datasets span multiple applications and tasks. For example, AssistGUI [61] focuses on automating GUI tasks using an actor-critic agent, serving as a benchmark for task-oriented GUI automation. VideoWebArena [74] evaluates long-horizon multimodal agents on web browsing tasks, emphasizing extended video context and web UI interactions. VideoGUI [104] compiles high-quality instructional screen recordings and introduces a hierarchical model for mapping visual observations to GUI actions. UI-Vision [128] provides a fine-grained desktop UI video benchmark with dense annotations for perception and interaction. Lastly, WorldGUI [201] increases task diversity by allowing arbitrary initial interface states for each task, challenging agents to handle varied starting conditions. These prior benchmarks primarily focus on close-ended tasks with predetermined goals, aiming to replicating expert demonstrations. In contrast, GUIDE targets open-ended GUI workflows with novice users, emphasizing understanding of user intent and context rather than step-by-step replication of actions. This shift toward user-centric evaluation fills a gap not covered by existing GUI video datasets that evaluate task completion or action prediction.

2.4.3 Video Question Answering

To enhance the comprehension of videos through question answering, a range of computational approaches has been explored. Some methods focus specifically on screencast tutorials, such as TutorialVQA [40] and PsTuts-VQA [202], which aim to support deeper understanding of software instruction videos. Broader-scale efforts leverage the extensive HowTo100M dataset to build large QA corpora, as seen in HowToVQA69M [181], iVQA [181], and How2QA [96]. However, many of these datasets rely on automatically generated questions, which may differ from the kinds of questions real users ask when learning from tutorials. To address this, some work has collected questions manually—either through crowdworkers generating questions from answer segments [40, 96, 181] or through domain experts crafting QA pairs [202]. Other efforts, such as YTCommentQA [189], have drawn questions from naturally occurring YouTube comments to better reflect authentic user information needs. Together, these datasets highlight diverse approaches to modeling the types of questions learners may have when engaging with instructional videos.

Chapter 3. VideoMix: Aggregating How-To Videos for Task-Oriented Learning

This chapter focuses on the first phase, the Exploration phase, where users learn from multiple tutorial videos. In this stage, the most useful contextual units are macro-level structures such as outcomes, approaches, and methods. This chapter has adapted and revised content from a paper at IUI 2025 [190]. All uses of “we”, “our”, and “us” in this chapter refer to the coauthors of the aforementioned paper.

3.1 Motivation and Contributions

How-to videos are a popular resource for people looking to learn new tasks (e.g., cooking a pasta dish or knitting a mitten) due to their abundance and the detailed, step-by-step instructions [188, 34]. When learning a task, people typically start with understanding the procedure and then applying it in their specific context [37, 5, 185]. This process involves gathering and processing information to construct an understanding of the task, followed by active engagement through execution and iterative learning via trial and error.

In the initial phase of learning, people often develop their understanding by watching or skimming through *multiple* videos. Watching multiple videos on the same topic can significantly enhance understanding of the task, by offering diverse perspectives and insights [92, 81]. This exposure allows users to learn about different methods, tips, or prerequisites, and select the approach that best fits their context. Additionally, learners can reference different videos to clarify any unclear points or to confirm the reliability of a specific method.

While this diversity provides such benefits, making sense of the loads of information in multiple videos is challenging. These videos are not curated, leaving the job of organizing and tailoring the information for the personal needs on the user. Navigating through numerous videos can be time-consuming, as most platforms are designed for viewing one video at a time, making related content fragmented and scattered. Moreover, since videos are not easy to skim, users must watch them sequentially, which can be inefficient. As a result, learners may end up watching only a few, potentially missing out on valuable information and knowledge. While systems like Surch [81] and RecipeScape [23] aggregate multiple procedures for a common task, they are specialized for specific domains (e.g., surgery) or primarily designed for analytical purposes, which often require domain expertise. Further exploration is needed to support learners in building a well-rounded understanding of tasks across a variety of domains.

To better understand why users watch multiple videos and what specific information they seek to gain from this process, we conducted a formative study in which we asked twelve participants to learn a task of their choice using how-to videos. We found that learners primarily look for four key aspects in the videos: **1) Outcomes**, to understand the possible results of the task and decide which outcome they prefer; **2) Requirements**, to identify the necessary tools or materials, and check whether certain tools are commonly used across videos; **3) Approaches and Methods**, to explore alternative approaches presented by various instructors and find the method that best suits their needs; **4) Details**, to gather additional insights, such as tips or know-how shared by different instructors. While participants recognized the value of watching multiple videos to gather this information, they noted the difficulty of tracking and organizing the information and the inefficiency of navigating between multiple videos.

Based on these findings, we developed VideoMix, a system that aggregates and organizes information from multiple how-to videos on a single task, helping users gain a holistic understanding of the task. VideoMix focuses on physical tasks with tangible outcomes, organizing videos into meaningful axes; outcomes, approaches, steps, methods, and details (Figure 5.3). Once the user inputs a task they want to learn, VideoMix identifies different outcome types (Figure 3.1B), and for each outcome type, VideoMix provides three different approaches to achieve the outcome: the standard (most commonly followed), the simplest (with the fewest steps), and the most complex (with additional steps) approaches (Figure 3.1C). Each approach is presented with the specific steps that make up the process, accompanied by a list of materials and tools used across the videos (Figure 3.1D, E). Once the user selects an approach they are interested in, they can explore different methods to achieve each step (Figure 3.2B). VideoMix provides video snippets demonstrating each method, along with useful tips or important details drawn from the videos (Figure 3.1C, E). To present potentially heterogeneous information from multiple videos in a coherent and digestable way, we integrate concise textual summaries with relevant video clips, enabling users to quickly digest and navigate the content.

To extract and generate this information, we designed a technical pipeline powered by a Vision-Language Model (VLM). Our pipeline processes a collection of videos to automatically extract key information such as outcome types, requirements, and step information along with relevant details from both the visual and verbal content of videos. A key component of our pipeline is the Dynamic Approach Identification (DAI) module, which captures different possible sequences of steps to achieve an intended outcome from a set of videos.

To evaluate VideoMix, we conducted a within-subjects study (N=12), where participants were asked to learn tasks that they had not done before, with our system and a conventional YouTube-like system. The results revealed that VideoMix helped participants gain an overall understanding of the task more efficiently, allowing them to tailor their learning experience by exploring approaches that matched their interests and suited their needs. Overall, VideoMix demonstrates the potential of task-based learning for videos, where videos are organized around a common task or goal, offering a concise yet comprehensive resource.

This paper presents the following contributions:

- A formative study that uncovers how users learn from multiple videos.
- VideoMix, a system that aggregates and presents information from multiple how-to videos on a task.
- An evaluation study that demonstrates the effectiveness of our system in task learning.

3.2 Formative Study

We conducted a formative study to gain insights into how users learn new tasks through multiple how-to videos and to understand the specific information they seek across these videos. In this section, we describe the methodology used and key findings identified from the study.

3.2.1 Method

We recruited 12 participants (6 male, 6 female, mean age=27.7, median=27) through online communities of academic institutions, who regularly watch how-to videos and often watch multiple videos to gain a comprehensive understanding of a task. All participants reported that they watch how-to videos of various domains such as cooking, painting, gardening, and assembly, at least 1-2 times per month.

To begin, we asked participants a few questions about their current practices on learning from how-to videos. We asked about the types of how-to videos they usually watch and asked them to describe their typical workflow, from watching the videos to following through with the task.

Next, participants were asked to select a topic or task they wanted to learn, ensuring it was a subject they had not previously learned or explored. Once the task was chosen, we conducted a think-aloud observation study. Participants were instructed to open YouTube, share their screen, and learn about the selected task as they would normally do. To simulate a realistic learning scenario, we asked them to imagine a setting where they had to learn about the task so that they could execute the task later. During the session, we observed how participants searched for videos, the specific information they sought, when and why they chose to look for another video and switch between them, and what information they gathered from each video. Participants were encouraged to think aloud about their thought process throughout the learning phase. We repeated the observation study with at least two tasks of the participant's choice, within a 45-minute timeframe.

Following the observation study, we conducted a semi-structured interview. We asked participants to describe the overall approach they used to learn the task, the types of information they found useful from different videos, the challenges they encountered, and the kind of support they would find helpful when navigating through multiple videos. The study was conducted online, and participants were compensated with a $30 USD Amazon gift card for the 1-hour session.

3.2.2 Findings

Current Workflows

All participants mentioned that when learning a task, they typically start by watching videos to understand the materials, processes, and techniques involved, forming a mental map before following the task. To watch videos, all participants began their video search with broad, general queries (e.g., *'how to make gnocchi'*), believing that these general queries would provide a better overview of the task and increase the chances of finding higher-quality videos, as a larger video pool is more likely to contain quality content. In contrast, they believed that more specific queries with personal contexts or constraints (e.g., *'how to make gnocchi without potato'*) might limit the search results. Additionally, since participants did not yet have an understanding of the task, they were often unsure about what specific details would be relevant to include in the search.

These broad queries yielded a large number of videos. All participants watched multiple videos when learning the task, and demonstrated two common behaviors for navigating through them. In the first behavior, demonstrated by five participants, they quickly scanned a list of videos and opened several videos in separate tabs, selecting those that aligned with their interests based on factors such as an appealing outcome, a title that matched their expectations (e.g., 'simple recipe'), or visual cues suggesting the video was of high quality. In the second behavior, observed in seven participants, they selected one video to watch at a time. Through watching that video, participants developed a better understanding of which personal constraints were relevant (e.g., not having a tool they needed), what specific outcome they wanted, or any knowledge gaps they needed to be clarified. They then accordingly refined their search queries for subsequent videos to become more specific and tailored to those needs.

Information Users Expect to See from Multiple Videos

Watching multiple videos allowed participants to get a broader understanding of the task and see various approaches and details that might not be covered in a single video. Below are the key pieces of information participants sought from multiple sources:

Outcomes: Participants quickly scanned video thumbnails and titles to grasp the specific outcomes of the task. For example, in learning how to "make gnocchi," they encountered variations like "cream gnocchi," "basil gnocchi," or "gnocchi soups." This allowed them to compare different end results and decide which version they wanted to pursue. Understanding these possible outcomes helped participants shape their goals and choose the appropriate approach.

Requirements: Participants also looked for the tools, materials, or ingredients used in the videos. By observing the requirements across multiple videos, they could identify commonly used items and ensure they had everything necessary to complete the task. This also allowed them to compare any unique items suggested by different instructors, helping them decide which tools or materials were essential.

Approaches and Methods: Participants explored various workflows presented in the videos, helping them identify both standard and alternative approaches. This comparison allowed them to understand the complexity of different methods and select one that best matched their skill level or specific context. Additionally, learning about different alternative methods provided flexibility and adaptability in their learning process.

Details: Participants appreciated the additional details that different videos provided, such as tips, tricks, or know-how. These insights added value to the learning experience, giving them more in-depth or practical knowledge that could enhance their understanding of the task.

Challenges

While participants found that watching multiple videos to be very beneficial to their learning, they also noted that the current process for using multiple videos is time-consuming and mentally demanding. They encountered the following challenges while trying to select, watch and organize information from multiple videos:

Search Results Lack Organization: The search queries always returned a large number of videos that weren't organized in a way participants could understand. As a result, participants found it difficult to select which video or videos to watch from the large set. For example, all participants primarily selected videos based on the outcome, which they determined from the search result titles and thumbnails. However, the search results were not organized by outcome; videos sharing a common outcome were scattered throughout the result list and participants had to exhaustively examine the list to comprehend all the possible outcomes for the task. Moreover, it was difficult for participants to gauge how videos sharing a common outcome differed. Better organization of the task videos based on the expected information types (Section 3.2.2) could help to reduce users' mental load.

Information Extraction Requires Watching Videos: Participants found it difficult to skim videos and spent a significant amount of time watching each video end-to-end in order to extract the information they wanted. For example, unless the original creators manually annotated the video or specified in the description box, participants often had no quick way to determine all the steps or ingredients used without watching the video through and risked missing important information while skimming. In contrast to video-only interfaces, past research has shown that mixed-media tutorials, which incorporate text, images and video together, are easier to skim and more effective at giving users a high-level overview of the task [34, 167].

No Easy Way to Compare and Consolidate Information Across Videos: As participants watch multiples video, they don't just want to gather information about each video independently. Instead, they were trying to form broader task insights which span multiple videos such as what the common approach is, which steps are not strictly necessary, or different methods to execute a single step. However, current video interfaces only support single video contexts; in order to watch multiple videos, participants had to open each video in a new tab and the videos were not aligned to each other in any way. This interface design made it difficult for participants to compare multiple videos and, as a result, participants spent considerable mental effort synthesizing and tracking these task insights. Additionally, participants also wanted to aggregate information across videos (e.g., all the tips and details from different instructors about a single step), but had no way of doing so in the current video browsing interface. Multiple participants expressed a desire for a system that could help them connect and consolidate the information from multiple videos more effectively.

3.2.3 Design Goals

From the formative study, we observed that watching multiple videos offered participants a more comprehensive understanding of a task, enriched with diverse instructions and insights. However, there was a need for a more efficient way to access and organize this information. Based on the study insights, we derive the following design goals for a multi-video system that is designed around a common task goal:

- DG1: Enable users to gain a comprehensive overview of possible outcomes and requirements for the task.
- DG2: Help users compare and navigate different approaches and methods to achieve the task.
- DG3: Provide easy access to detailed information, including relevant video snippets and key details shared across multiple videos.

3.3 VideoMix

Based on our design goals, we present VideoMix, a system that helps users gain a holistic understanding of a how-to task, by aggregating and organizing information extracted from multiple videos on the task.

3.3.1 System Interface

The system consists of an (1) Overview page (Figure 3.1) and (2) Details page (Figure 3.2). The overview page gives an overview of the task by organizing possible outcomes of the task, required materials and tools, and several approaches to achieve the task. Once the user selects an approach they are interested in, they see the steps that the approach involves. Once they click on a step, the system takes the user to the Details page, where users can see details for the step including multiple alternative methods and important tips, along with the corresponding video snippets.

Overview page

Once the user specifies the task they want to learn, VideoMix presents an overview of that task. First, it offers several possible outcomes (Figure 3.1B) for the task (e.g., for the task "Build a Desk," it shows options like ”Rustic Wooden Design,” ”Modern Sleek Design,” ”Functional Multi-purpose Desk,” or ”Standing Adjustable Desk”).

After the user selects a preferred outcome, VideoMix provides three different approaches (Figure 3.1C) to achieve it: the standard approach (the most commonly used across videos), the simplest approach (involving the fewest steps), and the most complex approach (involving the most steps). These approaches inform users of multiple ways to accomplish the task, varying in both commonness and complexity, while also providing flexible options tailored to their experience level and the amount of effort they wish to invest.

Once the user selects an approach, the system provides an overview of information gathered from multiple videos corresponding to that approach. First, a list of materials and tools used in the videos that follow the approach is provided (Figure 3.1D). Since not all items are used in every video, they are sorted by frequency of use—items appearing more often are highlighted with darker colors, making it easy for users to identify the most commonly used ones. Below the item list, the system displays step-by-step information for the approach, with each step labeled and briefly described (Figure 3.1E).

Details Page

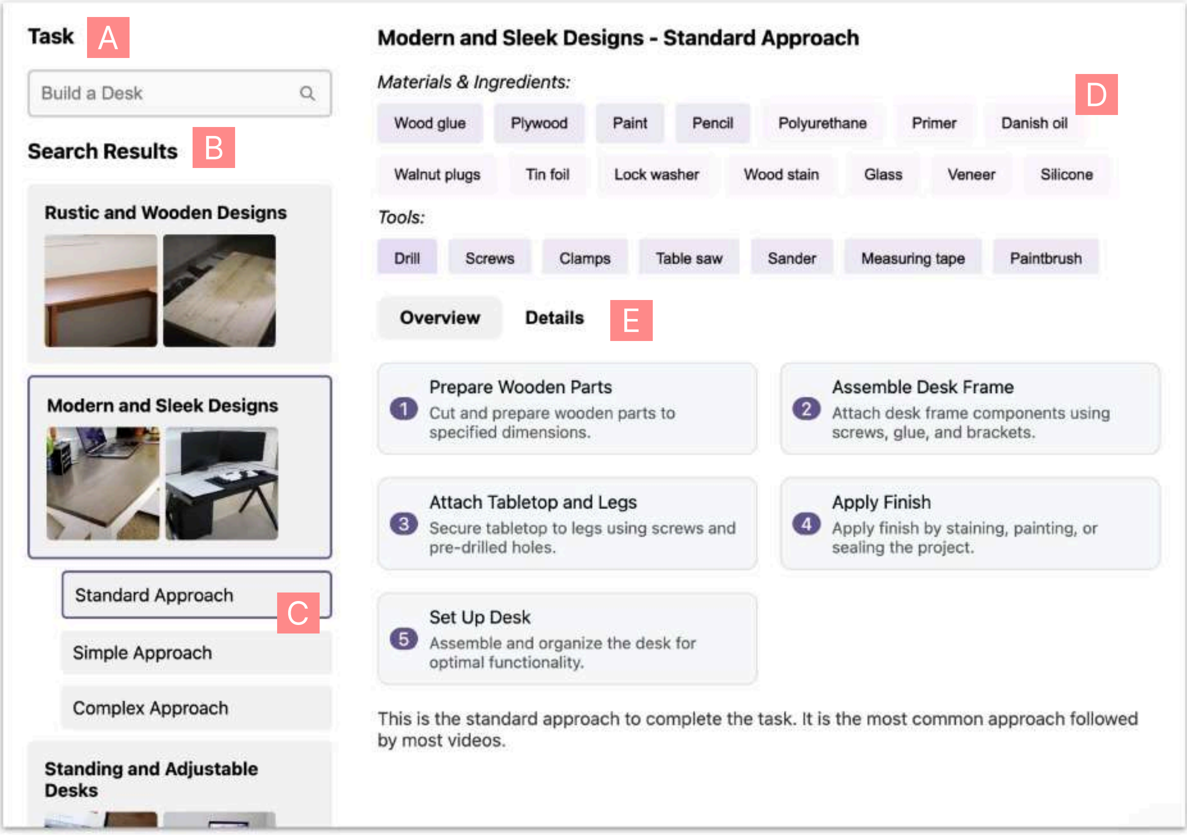
Once the user selects a step in an approach, they are presented with more in-depth information on the Details page(Figure 3.2). In this detailed view, VideoMix displays the step-by-step instructions previously shown in the Overview page, in a vertical format (Figure 3.2A). Here, each step can be expanded to reveal multiple variations or methods for accomplishing that step (Figure 3.2B). For example, for the step ”Cook meat and vegetables,” the user can choose between methods such as ”Using an Instant Pot,” ”Using a Rice Cooker,” or ”Using a Cast Iron Pot.”

Once the user selects a method, VideoMix presents video snippets corresponding to the chosen method (Figure 3.2C). These videos automatically play from the relevant start time and stop at the end of the segment, but users have the option to explore the video further by watching earlier or later parts to understand its context. On the right side of the video player, users can navigate between different video snippets, each accompanied by a brief summary (Figure 3.2D). This allows users to quickly understand the content of each snippet before selecting one to view, helping them explore different videos demonstrating the method. Below the video player, VideoMix provides useful tips and key information extracted from the video snippets to highlight important points or considerations for the selected method (Figure 3.2E).

As such, VideoMix enables users to gain a comprehensive understanding of the task by presenting information in a structured and hierarchical manner. This approach allows users to progressively learn about the task, revealing details as they delve into each outcome, approach, and step in depth.

3.3.2 Technical Pipeline

To provide the aggregated information from multiple how-to videos, we developed a pipeline that processes and extracts content in videos. Figure 5.3 illustrates the overall process. It begins by clustering videos into different sets based on their outcome and approach type. Each video set is then analyzed to extract more detailed information, such as steps and methods used. For the video dataset, we used HowTo100M [120], a large-scale collection of narrated how-to videos from YouTube. We downloaded the corresponding YouTube videos using youtube-dl [195], a command-line program for downloading videos from YouTube. We then obtained the video transcripts open-sourced by Han et al. [65], which were generated with sentence-level timestamps using WhisperX [11]. Each video is labeled with its task



Task

Build a Desk

Search Results

Rustic and Wooden Designs

Modern and Sleek Designs

Standard Approach

Simple Approach

Complex Approach

Standing and Adjustable Desks

Modern and Sleek Designs - Standard Approach

Materials & Ingredients:

Wood glue, Plywood, Paint, Pencil, Polyurethane, Primer, Danish oil, Walnut plugs, Tin foil, Lock washer, Wood stain, Glass, Veneer, Silicone

Tools:

Drill, Screws, Clamps, Table saw, Sander, Measuring tape, Paintbrush

Overview Details

1. Prepare Wooden Parts
   Cut and prepare wooden parts to specified dimensions.
2. Assemble Desk Frame
   Attach desk frame components using screws, glue, and brackets.
3. Attach Tabletop and Legs
   Secure tabletop to legs using screws and pre-drilled holes.
4. Apply Finish
   Apply finish by staining, painting, or sealing the project.
5. Set Up Desk
   Assemble and organize the desk for optimal functionality.

This is the standard approach to complete the task. It is the most common approach followed by most videos.

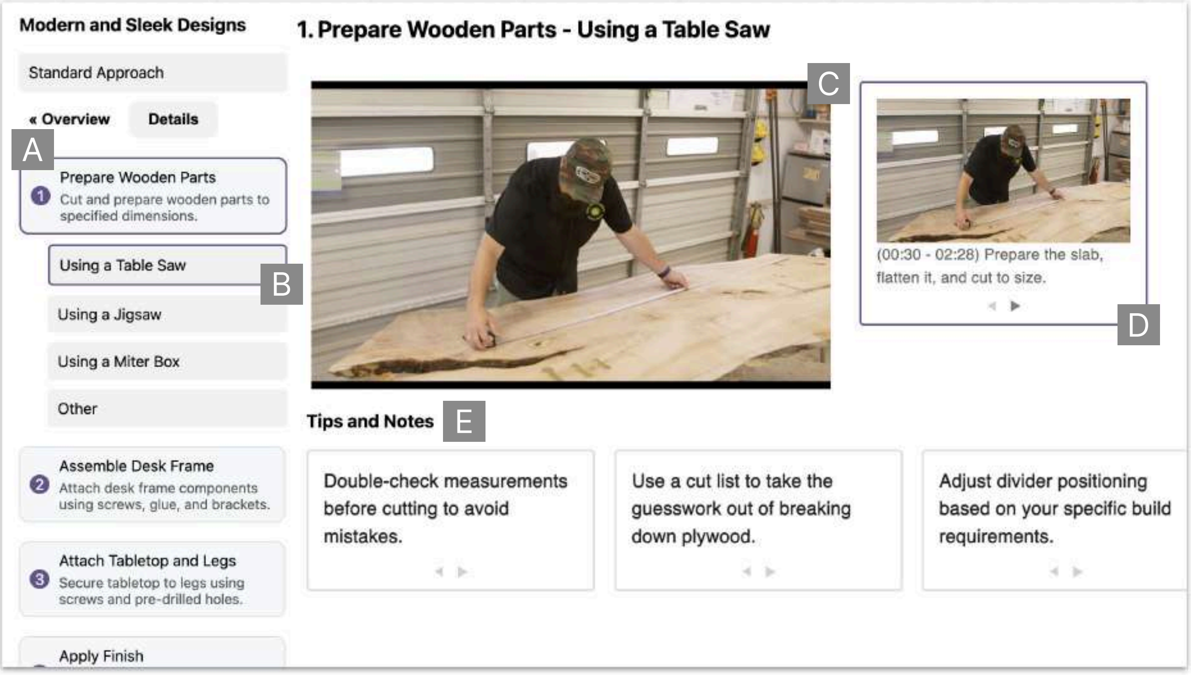
Figure 3.1: VideoMix interface on the Overview page for the task "Build a Desk". (A) Users begin by selecting the task they want to learn. (B) VideoMix then presents video search results categorized by outcome types. (C) For each outcome type, users can choose from standard, simple, or complex approaches. (D) Based on the chosen approach, VideoMix displays the necessary requirements, such as materials, ingredients, and tools. Finally, (E) users can see a list of steps and a brief description of each step that makes up the chosen approach.[2]

name (e.g., 'make gnocchi'), along with a broader category it belongs (e.g., 'Food and Entertaining').

Outcomes

To determine the different outcome types for a task, our pipeline operates in two phases: first, it extracts descriptions of each video's outcome and then it clusters these outcome descriptions into meaningful categories. In the first phase, we utilize both the visual content and transcripts. While transcripts provide verbal descriptions of the outcome [188], visuals can offer additional descriptive information that may not be explicitly mentioned. To estimate which video frames show the outcome, we provide GPT-4o with the full transcript and prompt it to extract only the segments that describe the outcome. We pick the video frames that correspond to these transcript segments as outcome frames, selecting one frame per second. We then input these outcome frames and the entire transcript into GPT-4o and prompt it to generate an outcome description. This phase yields an outcome description for each video in the task set.

[2] Screenshots of the outcome search results are from: youtu.be/CbJtZFXwxKY, youtu.be/Fnl1OwAAvEo, youtu.be/Z7x_Rvb_yjc, youtu.be/_v0fXgwcrpY (Creative Commons licensed).



1 Prepare Wooden Parts
Cut and prepare wooden parts to specified dimensions.

2 Assemble Desk Frame
Attach desk frame components using screws, glue, and brackets.

3 Attach Tabletop and Legs
Secure tabletop to legs using screws and pre-drilled holes.

(00:30 - 02:28) Prepare the slab, flatten it, and cut to size.

Double-check measurements before cutting to avoid mistakes.

Use a cut list to take the guesswork out of breaking down plywood.

Adjust divider positioning based on your specific build requirements.

Figure 3.2: VideoMix interface on the Details page for the task "Build a Desk". (A) The interface displays the list of steps for the chosen approach. (B) For each step, users can explore different methods, such as tools or techniques, to complete the step. (C) When a method is selected, VideoMix presents video snippets relevant to that method. (D) Users can easily switch between different videos for the selected method, with the corresponding time frame playing automatically. (E) Additionally, users can view tips and notes extracted from the videos.[4]

In the second phase, we cluster similar outcome descriptions together by outcome types. To extract the outcome type, we first prompt GPT-4o to identify two to four of the most salient themes from the list of video outcome descriptions. Each theme becomes an outcome type. We then cluster the videos around these outcome types by prompting GPT-4o to assign each video to exactly one outcome type using the video's outcome description. To provide representative images for each outcome type (Figure 3.1B), we randomly select two videos assigned to that type. We retrieve the outcome frame segments (identified in phase one) for each video and choose the middle frame of the last segment.

Steps and Approaches

To aggregate information from multiple videos sharing the same outcome, it is essential to understand possible sequences of steps that may vary across different videos [81]. We introduce a **Dynamic Approach Identification (DAI)** module, which iteratively identifies key steps across a set of videos, accounting for variations in the procedure. Instead of relying on a fixed taxonomy of steps for a task, our module adapts to a specific video pool (in our case, based on the outcome types of the task), and captures procedural differences within the set, ensuring comprehensive coverage of the task.

The DAI module, which is illustrated in Figure 3.3, begins by extracting steps directly from a video transcript and grounding each step in the corresponding transcript sentence indices using GPT-4o. Note that prior work [29] has demonstrated the feasibility and accuracy of using LLMs for step extraction with

[4] Source video: youtu.be/fv5bqBehcBc (Creative Commons licensed)

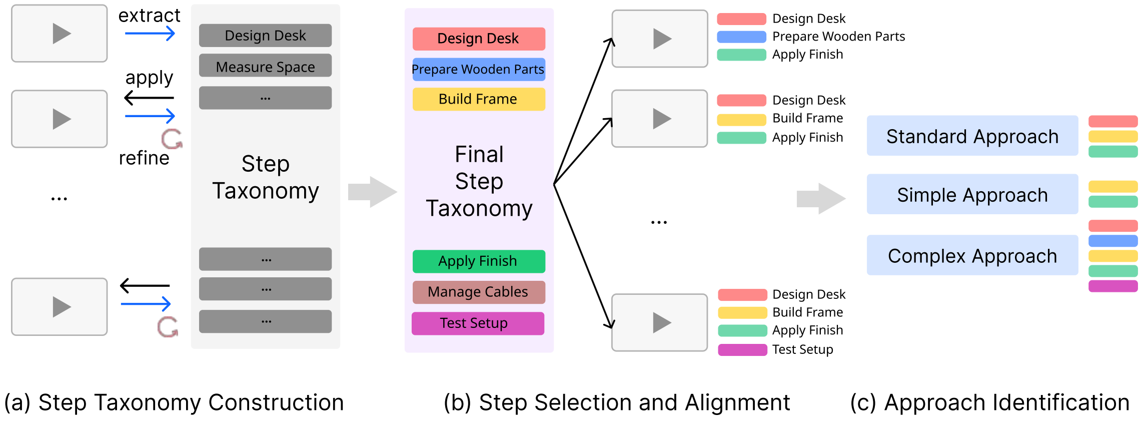


Figure 3.3: Illustration of our Dynamic Approach Identification (DAI) module, which captures a variety of approaches to accomplish a task. (a) The process begins by extracting step information from the first video using GPT-4o. This initial step taxonomy is then applied to the next video, where additional steps are identified, refining the taxonomy. This iterative process continues for all videos, progressively refining the step taxonomy with each comparison. (b) Once the final step taxonomy is established, it is reapplied to each video to detect relevant steps and align segments accordingly. Note that not all steps may be present in each video. (c) After extracting step information from each video using the common taxonomy, the system identifies standard, simple, and complex approaches based on the number of videos that follow each approach and the number of steps within each approach.

timestamps. The extracted step information is then applied to the next video to identify any previously unrecognized steps, adding those new steps to the set. This process is repeated iteratively, refining the step set until the entire video collection is covered. Once the final set of steps (i.e., the final step taxonomy) is derived, the system applies it to each video, selecting the steps present in the video with timestamp information for when each step occurs. This method allows us to capture each video's unique sequence of steps, which may or may not overlap with others.

Once the step information for each video is identified, our pipeline uses the information to determine three approaches: **Standard, Simple, and Complex**. The Standard approach refers to the typical sequence of steps most commonly followed across videos. The Simple approach refers to the sequence that involves the fewest steps, while the Complex approach consists of the largest number of steps. While there could be other ways to measure the complexity of an approach, we followed Merrill's suggestion [117] and used the number of steps as a measure, since it provides a quantifiable way to assess the effort required to complete the task. We execute the process of identifying steps and approaches for each outcome cluster, and the requirements are extracted per each approach. The standard approach is always captured, while the simplest and most complex approaches may not be, particularly if they overlap with the standard approach or if the number of videos following the simplest or most complex approaches is too low. In Section 3.4, we demonstrate how the DAI module effectively captures diverse and accurate approaches compared to existing baselines.

Object Requirements

Our pipeline also extracts the required objects (i.e., the materials, ingredients, and tools used) across all the videos belonging to the same approach. TutoAI [29] demonstrated that using LLMs to extract objects from transcripts is the most effective method for identifying items used in tutorial videos. To create a comprehensive list, we also capture visual frames at 5-second intervals from the entire video, and together with the entire transcript, prompt GPT-4o to extract the materials, ingredients, and tools used. After gathering this information for each video, we aggregate the results and calculate the frequency of each item across all the videos. To streamline the merging process, we instruct GPT-4o to exclude specific quantities or descriptors (e.g., stripping "pinch of salt" to be just "salt").

Methods and Details

Finally, our pipeline detects variations in the methods used for each step of an approach. For each step, we get the corresponding transcript segments from all the videos containing that step. We then prompt GPT-4o to identify the different variations in the methods described by transcript segments. To identify which of these method variations a video uses, we prompt GPT-4o with the video's step transcript and the method variations and ask it to pick which variation the step transcript describes. Finally, for each method, we prompt GPT-4o to extract useful tips or key information by providing a collection of transcript sentences specific to that method.

3.3.3 Implementation

The interface for VideoMix was developed using TypeScript, ReactJS, and CSS. The backend was implemented with Python scripts for video preprocessing. OpenAI's API was used for VLM components, specifically the GPT-4o-2024-05-13 model [135] with a temperature setting of 0 for all components. To generate structured outputs, we employed Function Calling [134] in OpenAI's API. Note that we used GPT-4o to process video frames and transcripts for a robust and scalable solution for handling long-form videos. We did not use video foundation models due to their limited context windows, which make processing lengthy videos challenging without losing details. Future improvements in video foundation models, such as larger context windows and lower costs, could make long-form video processing more efficient and practical for our pipeline.

3.4 Technical Evaluation

We evaluated the Dynamic Approach Identification (DAI) module primarily, as it is the core component of our pipeline for identifying diverse approaches and methods across multiple videos. We aimed to test two hypotheses: 1) Our pipeline-generated step taxonomy will provide as accurate step information as predefined taxonomies; 2) Our pipeline-generated step taxonomy will better capture the diversity and variation within a task compared to predefined taxonomies.

3.4.1 Task Selection

To evaluate our hypotheses, we selected six tasks from the HowTo100M dataset, with two from the 'Hobbies and Crafts' category and four from the 'Food and Entertaining' category. The chosen tasks are: *Build a Desk* (95 videos), *Build a Bookshelf* (58 videos), *Make Chicken Cacciatore* (92 videos), *Make*

	Accuracy (1-7)			Coverage (0-10)
	Relevancy	Logical Flow	Completeness	
Baseline	5.88 ± 1.19	5.58 ± 1.56	4.50 ± 1.67	5.80 ± 2.24
VideoMix	5.42 ± 1.36	5.52 ± 1.38	4.42 ± 1.53	(1) 7.05 ± 2.13 (*) (2) 7.96 ± 1.83 (**)

Table 3.1: Results of the technical evaluation of our DAI module. Our pipeline maintained step accuracy across Relevancy, Logical Flow, and Completeness (with no statistically significant differences), while capturing a significantly more diverse range of possible approaches, both (1) when considering only the approaches and (2) across all outcome types (*: p¡0.05, **: p¡0.01).

Jambalaya (66 videos), *Make Shrimp Cocktail* (86 videos), and *Make Bannock* (90 videos). The tasks were selected based on the following criteria: 1) We focused on physical tasks with tangible outcomes, rather than fixing or using products [188]. This was to ensure diversity in information, such as outcome types and requirements. 2) The task must have a predefined step taxonomy available in existing datasets (e.g., HT-Step [2], CrossTask [208]) to allow for comparison. 3) The task must include at least 50 videos to ensure diversity. For comparison, we used HT-Step and CrossTask as baseline datasets, since both are also based on HowTo100M. The step taxonomies in these datasets are human-annotated, grounded in WikiHow [176], a popular website for how-to instructional articles.

3.4.2 Method

We recruited external evaluators through Prolific [150], who are familiar with the selected tasks. In total, 24 evaluators were recruited, with 4 evaluators assigned to evaluate each of the 6 tasks. To ensure expertise, we required evaluators to self-report having performed the task at least once and to know at least two approaches to completing it. Evaluators were asked to rate the step information derived from both the baseline predefined step taxonomies and our pipeline-generated steps for the same video tasks, where the order of the condition was counterbalanced.

The evaluation focused on two main criteria following our hypotheses: accuracy and coverage. For accuracy, evaluators rated the step information based on the following criteria using a 7-point Likert scale:

- Relevancy: How relevant is each step to achieving the overall task goal?
- Logical Flow: How logical and coherent is the progression of steps in the sequence?
- Completeness: How complete is the sequence in covering all necessary steps to achieve the task?

For the baseline, evaluators were presented with the predefined step taxonomies, but we summarized each step into a concise step name to ensure consistency with the format of our pipeline-generated steps. For our pipeline-generated taxonomies, evaluators were provided with the 'standard' approach for each outcome type. For coverage, evaluators answered the following question on a scale of 0 to 10 where 0 indicates no coverage and 10 means a full, 100% coverage:

- To what extent does this sequence represent or cover all the possible ways to achieve the task?

In this case, evaluators were provided not only with the standard approach but also with simple and complex approaches for each cluster, if available, in the pipeline-generated taxonomies. Evaluators were compensated $5 USD for each task they evaluated, which took approximately 15 minutes.

3.4.3 Results

Overall, our pipeline maintained step accuracy while capturing a more diverse range (80%) of possible approaches compared to the baseline (58%). For accuracy, when evaluated on three key aspects—Relevancy, Logical Flow, and Completeness—using a 7-point Likert scale, there were no statistically significant differences between the steps generated by our pipeline and those annotated by humans. (Table 3.1, Relevancy: μ=5.88, σ=1.19 vs. μ=5.42, σ=1.36; Z=1.5, p=0.13, Logical Flow: μ=5.58, σ=1.56 vs. μ=5.52, σ=1.38; Z=0.37, p=0.71, Completeness: μ=4.5, σ=1.67 vs. μ=4.42, σ=1.53; Z=0.47, p=0.64). Note that each condition was evaluated according to its intended outcome. The baseline involved the general task (e.g., building a desk), while our pipeline was tested on specific outcomes (e.g., building a standing adjustable desk). These results indicate that our pipeline can generate steps with a level of quality comparable to human-annotated steps, even when addressing more specific tasks.

In terms of coverage, the steps generated by our pipeline captured a significantly greater range of possible approaches to completing the task, as rated on a scale from 0 to 10, (0 being 0% and 10 being 100%). Compared to the baseline steps, our pipeline captured a more diverse range of approaches, even when considering only the approaches (i.e., Standard, Simple, and Complex) for each intended outcome type. (Table 3.1, μ=5.8, σ=2.24 vs. μ=7.05, σ=2.13; Z=-2.16, p¡0.05). When aggregating these approaches across all outcome types, the coverage increased significantly from 58% to 80%, with an average of 3.5 outcome types per task (Table 3.1, μ=5.8, σ=2.24 vs. μ=7.96, σ=1.83; Z=-3.37, p¡0.01). These results demonstrate that our pipeline, which detects step information across various outcome types and approaches, captures significantly more diverse ways to achieve a task. All statistical significance was measured using the Wilcoxon Rank-Sum Test.

3.5 User Study

We conducted a within-subjects user study to evaluate VideoMix against a baseline YouTube-like system, a platform most users are familiar with for watching how-to videos. The primary goal of the study was to assess the effectiveness of VideoMix in enhancing users' overall understanding of tasks, and to explore how users would use VideoMix and how it impacts their learning experience.

3.5.1 Participants and Apparatus

We recruited 12 participants (4 male, 8 female, mean age=25.3, median=25.5) through an online community at our academic institution, those who regularly watch how-to videos and often watch multiple videos to learn a specific task. For the study, we selected 4 tasks from those used in our pipeline evaluation: two from the 'Hobbies and Crafts' category (*Build a Desk, Build a Bookshelf*), and two from the 'Food and Entertaining' category (*Make Chicken Cacciatore, Make Jambalaya*). We randomly selected two tasks for each participant, one for VideoMix and the other for a baseline system. Since our study involved learning tasks, we ensured that none of the participants had prior experience with the tasks they would be learning during the session.

For a fair comparison between VideoMix and baseline, we built a baseline system similar to YouTube, but with a limited set of videos available in VideoMix. Participants were provided with a list of videos in the main feed, where they could click to watch each video along with its title and description sourced from the original YouTube video.

3.5.2 Study Procedure

The study was conducted online through a Zoom meeting. Participants were first given an overview of the study, including the two tasks they would be learning during the study. They were then instructed to use either VideoMix or the baseline system to learn about an assigned task. Participants were asked to imagine they would later be performing the task on their own, and that their current goal was to study the task, gather as much information as possible to prepare for it.

We provided a brief tutorial on how the assigned system worked, and participants were given 15-20 minutes to explore and learn about the task using the system. They were encouraged to think aloud, sharing their thoughts and decision-making process as they use the systems. After completing one session, participants switched to the other system, and the same process was repeated. The order of tasks and systems used were counterbalanced across participants. Following each session, we conducted a questionnaire to assess participants' perceived understanding of the task, perceived usefulness of each feature (in the VideoMix condition only), and cognitive load using measures from NASA-TLX (*Mental Demand, Frustration, Effort, Performance*) [66]. All responses were on a 7-point Likert scale. Finally, we conducted semi-structured interviews to understand their strategies used in each system and gather qualitative feedback on VideoMix. The study lasted 1 hour, and participants were compensated with a $30 USD Amazon gift card.

3.5.3 Results

Overall, participants found VideoMix to be more helpful in understanding the task compared to the baseline. Below, we provide a detailed report of the study's findings. For all measures, we first conducted a Shapiro-Wilk test to determine data normality, and then used a paired t-test (if parametric) and a Wilcoxon signed-rank test (if non-parametric).

Enhanced Overall Understanding

Participants reported a significantly better understanding of the tasks when using VideoMix compared to the baseline (Figure 3.4). They felt more successful in learning about the task (μ=4.83, σ=1.4 vs. μ=5.75, σ=0.83; t=-2.42, p¡0.05) and more efficient in the learning process (μ=4.17, σ=1.9 vs. μ=5.75, σ=0.92; W=7.0, p¡0.05). Participants appreciated how VideoMix provided a comprehensive overview of the task, allowing them to grasp the entire scope at a glance. For instance, P2 noted, *"With VideoMix, I could see the overall process involved in the task and get a general understanding immediately. I could figure out possible outcomes, required materials, and overall process, which would have taken a long time to find on YouTube, where videos are scattered."*

VideoMix significantly streamlined the process of acquiring task-related information compared to the baseline. With the baseline, participants typically relied on thumbnails to identify the outcome or titles to see the approach they wanted (e.g., 'simple recipe'). After selecting a video, they would check the description box in hopes of finding a list of ingredients or basic step-by-step instructions, but this information was not always available. In contrast, VideoMix offered organized information upfront, saving

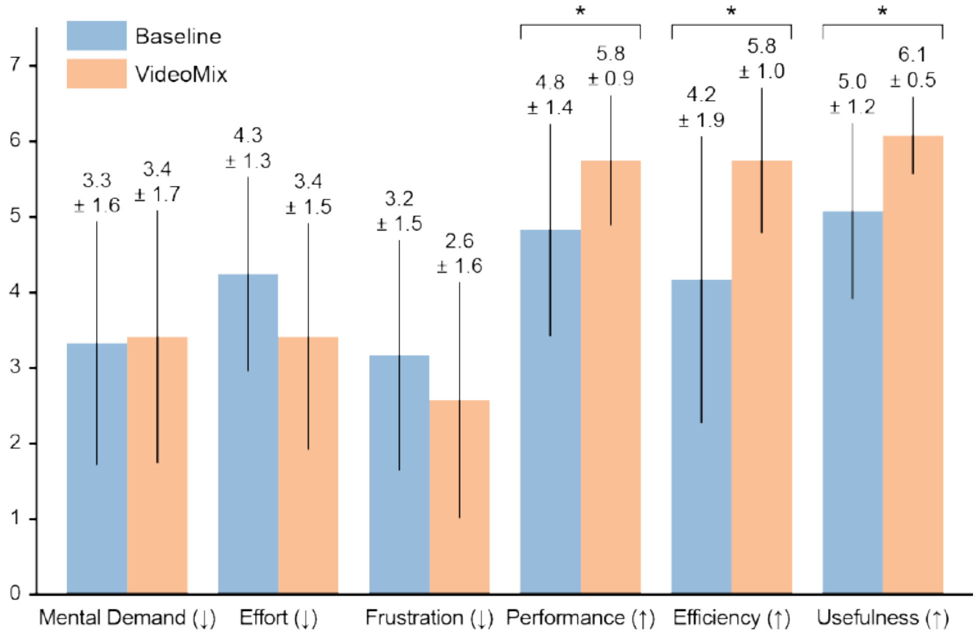


Figure 3.4: Participants felt they were more successful and efficient with VideoMix, and found VideoMix to be more useful when learning about the task compared to the baseline. There were no statistically significant differences in mental demand, effort, and frustration (*: p¡0.05).

participants considerable time. For example, P6 selected the standard approach in VideoMix because he wanted to learn something basic, whereas on the baseline, he had to watch multiple videos and compare processes to identify the original standard recipe. He also mentioned, *"It's nice because the ingredients are written out, so you can just look at them and prepare everything right away."* While VideoMix presented information from an average of 77.8 videos per task, participants watched only 2.6 videos on average using the baseline system within the given study time.

Overall, participants rated VideoMix to be significantly more useful for gaining an overall understanding of the task compared to the baseline system (μ=5.08, σ=1.16 vs. μ=6.08, σ=0.49; t=-2.87, p¡0.05). 10 out of 12 participants mentioned they would prefer VideoMix to baseline when understanding a task. However, there were no statistically significant differences in mental load, frustration, or effort during the learning process.

Tailored Learning Experience

VideoMix organizes instructional content from multiple videos into a hierarchical structure based on outcome, approach, and method employed. This allowed participants to efficiently focus on instructions that best suited their specific needs and context.

First, the outcome types helped participants narrow their focus to what they were most interested in learning. After exploring various outcome choices, participants developed a clear preference based on either personal tastes (e.g., *Jambalaya with Chicken and Sausage* vs. *Vegan or Low-Carb Jambalaya*) or estimated proficiency level (e.g., *Modern and Sleek Design Desk* vs. *Functional and Multi-purpose Design Desk*).

Next, the different approaches enabled participants to choose learning pathways that matched their experience level. Most participants, being new to the task, looked for simple or standard methods. P8

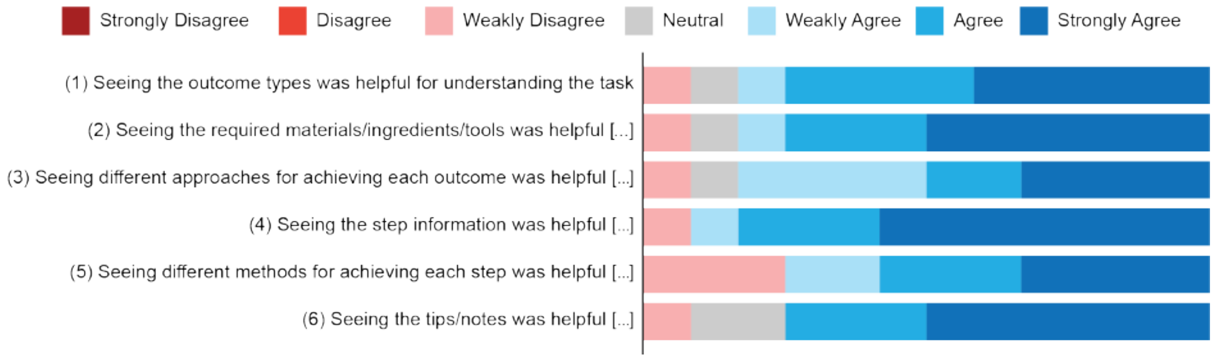


Figure 3.5: Participants' ratings on the usefulness of each information piece in understanding the task. Overall, they found the information provided by VideoMix—including outcome types, requirements, different approaches, step details, methods, and tips and notes—to be helpful in gaining a better understanding of the task.

remarked, *"It was easy to have a clear criterion, whereas on YouTube, I had to guess content from thumbnails and titles. Even if the first video I watched had a unique approach, I might have assumed it was the original recipe for Jambalaya. I would have spent much more time than I did with VideoMix to find a recipe that fit my situation."* P8 only realized that one of the three videos she watched on YouTube matched her beginner level after viewing all three.

Finally, the variety of methods allowed participants to focus on instructions that aligned with their available tools and ingredients. For example, P3 said, *"It was helpful to see different methods because I don't have an oven, so I looked at the `Using Stove` or `Using Pot` methods instead of `Using Oven`."* In contrast, finding a video that fit their context on YouTube was often more challenging. P7 noted, *"As I watched the video, I was concerned that I didn't have the right equipment or materials used in the video, and thought I'd probably need to search for another one."* In summary, VideoMix enabled participants to learn more effectively by providing clear, relevant options that could be tailored to their specific preferences and resources.

Knowledge Acquisition By Multi-Video Comparison

VideoMix allows users to easily navigate between videos showcasing the same method within a step (Figure 3.2D). By comparing multiple segments, participants gained a deeper understanding of the methods. For example, P4 said, *"Even though both video segments I watched were all about using wood glue, one video showed how to apply it while the other explained when to use it. This helped me understand the step better."* Similarly, P1 initially didn't know what Leger Boards were in Building a Bookshelf when only watched a single video segment, but learned what they are after watching multiple segments using them.

Participants also picked up key information about requirements and techniques. For example, P8 said, *"I saw that celery and garlic were used across all standard approaches of different outcomes, so I realized they are key ingredients."* P9 highlighted how different methods offered contrasting tips, saying, *"For the Instant Pot, tips suggested adding vegetables first, while for the rice cooker, they recommended adding meat first. The order seems important based on the tool you're using."* The ability to compare multiple perspectives within the same task participants' understanding, offering a more comprehensive learning experience.

Further Improvements for VideoMix

While participants found VideoMix to be an effective tool for learning new tasks through videos, they noted suggestions on how VideoMix can be further improved. First, they mentioned the discontinuous nature of the segmented videos throughout the steps could hinder the learning process. For example, P1 said, *"When I clicked a next step and a segment from a new video was shown, it took me a while to understand the context of the video."* Participants expressed a desire to see a continuous video, while having the information VideoMix offers. P2 said, *"It would be great if I could select one main video, and see additional details not covered in that video through VideoMix."* A potential improvement could be a hybrid format, where users first watch a full video, and what VideoMix currently provides is organized around that primary video.

Participants also suggested ideas on how methods are presented. For example, P10 suggested sorting the methods by commonness, similar to how requirements are organized or how VideoMix shows the 'Standard Approach' (as we do for the approaches). P12 wished to see the outcome of each video segment to better choose which method to follow, similar to how VideoMix shows different outcome types for the task on the overview page. This feedback suggests that the hierarchical structure VideoMix uses to organize task-level information could be re-applied at the step level, providing more detailed information.

3.6 Discussion and Future Work

In this paper, we present VideoMix, a system that aggregates multiple how-to videos to provide a comprehensive understanding of a task. We discuss how it supports task learning, considerations for designing multi-video systems, the incorporation of the hierarchical nature of tasks, and potential directions for future work.

Supporting Task Learning: from Understanding to Following

VideoMix is designed to facilitate task learning by helping users synthesize multiple videos, enabling a better understanding of the task. This aligns with a key search intention in Information Retrieval [155], which emphasizes learning domain knowledge. While VideoMix is primarily intended to assist users in the understanding phase before they move on to task execution, 7 out of 12 participants expressed interest in using it throughout the task-following phase as well.

Participants highlighted several benefits of VideoMix in task following: it presents various methods together, allowing users to choose their preferred approach as they follow the task without searching through multiple videos (P2); its mixed-media format with text makes following instructions easier (P6); and the segmented steps enable users to quickly revisit specific parts of the process (P11).

However, other participants preferred YouTube for task *following*, citing the importance of consistency and flow. As P9 noted, *"Mixing two different recipes is generally not a good idea."* While a few participants could identify the same video across different steps by recognizing the background or demonstrator, it remains important to support the tracking of a cohesive procedure within a single video, especially in the following phase.

To better support the full learning cycle—from understanding to following—we envision a system that allows users to explore various methods (as VideoMix currently does), then select specific videos for following, while maintaining easy access to overview information [81]. To better support the following phase, we suggest features like real-time prompting or interactive search to address the users' more specific needs as they progress through the task. Additionally, while VideoMix offers some customization by providing a list of tools for each approach and outcome type, or methods specifying tool usage, allowing users to retrieve videos based on selected tools or choose the level of detail they want to explore could further improve the customization experience.

Designing Multi-Video Systems

VideoMix organizes information from multiple videos to provide a comprehensive understanding of tasks. Instead of treating videos as the primary object, VideoMix treats the task itself as the first-class object, with multiple videos structured around it. Thus, the basic unit is a video segment (i.e., part of a video), which is then organized around a task.

Designing a multi-video interface around video segments has both advantages and challenges. On the positive side, splitting content by steps made it easier to digest, and multiple demonstrations for each step enhanced learning (Section 3.5.3). However, users could feel a sense of discontinuity between segments and sometimes lack the broader context of the full video (Section 3.5.3). To address this, a multi-video interface should ensure that enough context is provided and consider strategies to maintain continuity, such as using a common voice-over, visual connectors, or a consistent theme across videos.

Another challenge is managing the extensive amount of information drawn from multiple videos, which may feel overwhelming to some users. Two out of 12 participants who preferred YouTube over VideoMix appreciated its ability to present diverse methods at a glance but found the overall information density to be excessive. While VideoMix aims to reduce the time required to learn viable methods through structured presentation—particularly for tasks with high variability—this comes with trade-offs. Curating information may limit certain details as well, and it is essential to balance organization with user agency in the exploratory search process.

Lastly, it would be interesting to explore how a multi-video interface might reshape user engagement, especially in interactions typically supported by traditional video-centered platforms, such as commenting, liking, or sharing. Investigating how these interactions can be adapted to a multi-video interface, as well as identifying potential new interactions unique to this interface, would be an interesting avenue for future research.

Incorporating Hierarchical Nature of Tasks

How-to videos naturally contain hierarchical information [196]. Tasks often consist of multiple sub-tasks or steps, each of which could be a task on its own. For example, in the task of making an Eggs Benedict, one of the steps might involve poaching an egg, where there could be videos solely about it.

This hierarchical structure presents an opportunity for VideoMix to further enhance learning by extending its current task-level organization to a more granular, step-level structure. Just as VideoMix organizes information by outcome, requirements, and approaches at the task level, the same principle could be applied recursively at the step level (as briefly discussed in Section 3.5.3). For instance, the step of poaching an egg could be broken down into sub-steps such as preparing water, cracking the egg, and cooking the egg, where there could be multiple variations within each sub-step. This approach would allow users to delve deeper into specific areas of interest, fostering a more flexible and personalized learning experience. By supporting this recursive exploration, users could not only learn how to complete a task like making an Eggs Benedict but also master individual skills, like poaching eggs, that could be applied in a wide range of other contexts, supporting a flexible and infinite journey of learning.

Limitations and Future Work

Our pipeline only requires videos to have narration, as it relies on spoken content to extract task steps and details. As long as videos are accessible and can be transcribed using ASR, our approach remains applicable. However, a key limitation of VideoMix is its dependence on the quality and quantity of the source videos. Since the system compiles content from various videos, the clarity of the presenter's instructions and the logical flow of the content can significantly affect its performance. In particular, VideoMix relies heavily on transcripts for extracting steps and methods, making clear and well-structured narration essential. If a video lacks coherence or clarity, the system may struggle to extract accurate and meaningful information.

In terms of quantity, our system may not provide as comprehensive an overview when the available videos are limited (e.g., only 10 videos on a given task). In such cases, we could consider expanding the search to include more videos (e.g., similar methods used in different tasks) or incorporating other tutorial resources, such as text-based materials. Similarly, while we demonstrated VideoMix based on videos selected from the HowTo100M dataset [120], expanding the video pool through additional crawling would allow VideoMix to offer a richer and more diverse set of instructions. By refining search queries to capture more hierarchical videos (e.g., searching for specific outcome clusters or individual methods), the system could provide a broader range of instructional content. We believe that as VideoMix processes more videos, its comprehensiveness and ability to support users will improve.

Additionally, VideoMix has primarily been tested on tasks involving the creation of physical objects, which typically feature well-defined steps and clear visual and verbal cues. However, extending VideoMix to other types of tasks—such as digital tasks like Photoshop editing or guitar tutorials—may introduce new challenges. For example, tasks like guitar tutorials may require a different structure that emphasizes progressive skill building rather than multiple methods to achieve the same step. They may also rely more heavily on subtle nuances such as hand placement, tone, or timing, which are difficult to capture solely through transcripts. Beyond how-to tasks, there is potential for VideoMix to be applied to other domains, such as organizing interview videos by specific questions or themes. By structuring interviews around common topics across multiple videos, the system could provide users with a comprehensive view of diverse perspectives. This approach could also be extended to educational content, where VideoMix could organize lectures by subtopics, offering a clearer, more structured learning path for users.

3.7 Conclusion

This paper presents VideoMix, a system that helps users gain a comprehensive understanding of how-to tasks by aggregating information from multiple tutorial videos. We demonstrated that VideoMix enables users to explore different methods, materials, and outcomes more easily, leading to a better understanding of a task. Our work highlights the potential of a task-oriented, multi-video approach to support users in task learning. As online tutorials and video content continue to grow, our system provides an important step forward in improving how people learn from them.

Chapter 4. Beyond Instructions: A Taxonomy of Information Types in How-to Videos

This chapter focuses on the second phase, the Comprehension phase, where learners begin narrowing down to a specific video. In this stage, information types such as justifications and tips serve as useful contextual units. This chapter has adapted and revised content from a paper at CHI 2023 [188]. All uses of "we", "our", and "us" in this chapter refer to the coauthors of the aforementioned paper.

4.1 Motivation and Contributions

How-to videos provide procedural information about performing tasks such as cooking, makeup, and crafting. They explain how to perform a task by visually demonstrating workflows while providing verbal explanations. Due to their detailed explanations, how-to videos have been a popular source of help when performing a task [80, 34].

There is diverse information beyond instructions intertwined in how-to videos. In addition to instructional information about how to perform each step, instructors share their strategies for choosing supplies [35] or give additional commentary [167]. They also share their personal tips or pitfalls [32], or even ideas not directly related to the task, such as greetings or jokes [119].

From the sea of information, each user requires different information that caters to their specific purpose or situation of watching videos. Depending on their needs, users might want to see only relevant instructions [80], ingredients or tools used, or check the final outcome of a video [127]. To help users find the content of interest, the most common approach has been to enable chapter-based navigation where it segments the video into coherent subtopics [192, 167, 175, 56, 84, 149, 127, 141, 34]. It allows users to navigate videos based on subtopics in videos and locate a section of interest.

However, the diverse information within a video is scattered throughout, making it difficult for users to identify information that meets their needs. Even a chapter contains various types of information. Moreover, the diverse kinds of information are intertwined in no particular order. The author may proceed to offer their rationale, describe intermediate outcomes, or even promote their channels in the middle of giving instructions at any part of the video. The unpredictability of a video's structure makes it even more difficult for users to retrieve the information they need.

We propose that a comprehensive taxonomy that identifies and categorizes the types of information shared in how-to videos can serve as a foundation for supporting users in navigating videos. It provides a structural basis for analyzing and understanding users' navigational behavior. It facilitates the understanding of useful information types for different user needs arising from distinct settings such as the purpose of watching or the domain of the video. Understanding how users leverage information types to navigate videos will ultimately lead to better designs of video navigation systems that suit users' needs.

To this end, we investigated verbal utterances from how-to videos to identify and organize information types in how-to videos. We focused on verbal utterances as the primary source of information because they often contain explicit explanations of what instructors demonstrate [35, 120], sometimes giving additional information that is not visually available. Thus, we presume that verbal information would cover a wide range of information delivered in how-to videos.

To construct the taxonomy, we selected 120 videos from the HowTo100M dataset, a large-scale dataset of narrated how-to videos that covers 12 different genres (e.g., Cooking, Arts, Sports) [120]. We performed an iterative open coding of 4k sentences from 48 videos to generate a taxonomy of information types in how-to videos. From the analysis, 21 information types emerged under 8 categories: *Greeting*, *Overview*, *Method*, *Supplementary*, *Explanation*, *Description*, *Conclusion*, and *Miscellaneous*.

To validate the taxonomy, we applied the taxonomy to a total of 120 how-to videos containing 9.9k sentences which we contribute as a dataset, HTM-Type[1]. From the analysis of the dataset, we found that `Method`, the core information required to complete the task, makes up 47.5% of the video time on average. We also found that the task type (i.e., Creating, Fixing, or Using) and narration style (i.e., Real-time or Dubbing) affect the distribution of information types, and that certain categories have a temporal tendency.

After creating and validating the taxonomy, we demonstrate the utility of the taxonomy in both analyzing users' navigational behavior and supporting their navigation in how-to videos. We first show how our taxonomy can serve as an analytical framework for existing video systems that were built to support video navigation. We observed that the systems utilized different information types to meet users' specific needs. To further investigate how users leverage information types in various navigation tasks, we built a research probe that enables users to navigate using the information types within the video. Through a user study with nine participants, we observed that the participants effectively used different information types for finding specific information needed to perform each of the Search, Summarize, and Follow tasks. We further discuss how our taxonomy can enable a number of applications in video authoring, viewing, and analysis.

This paper makes the following main contributions:

- A taxonomy of information types in how-to videos
- HTM-Type, a dataset of 9.9k sentences from 120 videos labeled according to the taxonomy
- Empirical findings on how people use information types in navigating videos

4.2 Taxonomy of Information Types in How-to Videos

To examine the diverse information types present in how-to videos, we conducted a content analysis on how-to videos. The goal of our analysis was to identify information types, which are the intent behind the units of content in videos. We chose verbal utterances as the primary source of information in our research scope. This is because instructors often explicitly describe the visual content such as what they are doing or what is happening [35, 120], sometimes giving additional information that is not visually available. However, we also considered visual information as an additional factor to take context into account, because sometimes it is hard to identify the type of information the instructor is delivering just from the textual description. For example, when the instructor uses pronouns such as *"it"* or *"this"*, it is hard to know what they are referring to (e.g., tool, method, or situation). Also, it is hard to tell if a sentence is a joke or an instruction without watching the actual situation (e.g., *"What do you do with the half you have leftover? Dip it in some hummus, of course."*). Below we describe our approach to generating the taxonomy and present the results.

[1] videomap.kixlab.org

4.2.1 Methods

Data Collection

We selected videos from the HowTo100M dataset, a large-scale dataset of narrated how-to videos [120]. The dataset covers 12 different genres of how-to videos, organized according to the categories in WikiHow [176]: *Arts and Entertainment, Cars and Other Vehicles, Computers and Electronics, Education and Communications, Food and Entertaining, Health, Hobbies and Crafts, Holidays and Traditions, Home and Garden, Personal Care and Style, Pets and Animals, and Sports and Fitness.* To ensure that we cover a wide range of topics, we selected 10 videos from each of the 12 genres, resulting in 120 videos in total.

We first filtered for videos that were longer than 5 minutes to ensure a sufficient amount of content and that were produced within the last five years (that is, 2017 or later) to reflect the most recent and relevant production trends in how-to videos. To acquire the duration and publication date of the videos, we used youtube-dl [195], open-source software for downloading videos and the related metadata. Then, we went through each of the filtered videos and selected 10 videos from each of the 12 genres that 1) are narrated in English, 2) have one person demonstrating, and 3) are in the scope of "how-to videos", namely explaining how to get a task done[2]. After selecting the videos, we transcribed them using Microsoft Azure Speech-to-text API [10], which transcribes the spoken language in videos with timestamps of each word using Automatic Speech Recognition. Then, we used a BERT-based punctuation model [133] to split the transcripts into sentences.

Constructing the Taxonomy

After selecting the videos, three of the authors performed an iterative open coding for the content analysis of the videos. We individually coded each sentence based on the type they believed it to be conveying. We watched the videos while identifying the types to make sure we incorporated the exact context of each sentence and clarify any errors in the transcript. Also, we split a sentence if it contained two or more information types so that each sentence only contains one information type. The total number of split sentences was around 1% of all sentences. Then, we resolved each conflict through a discussion between the three authors and merged the codes every six videos.

To ensure the validity of our taxonomy, we set two criteria for its construction following the practice in taxonomy development [130]: (1) All elements in the taxonomy should be mutually exclusive (i.e., no overlapping between elements) and (2) the taxonomy should be collectively exhaustive (i.e. cover everything). First, to verify that all elements are mutually exclusive, we convened every session to discuss the discovered information types and whether they were mutually exclusive or could be divided into smaller parts or merged. If there were any ambiguous sentences that could be interpreted as multiple types, we handled those cases by figuring out what factors caused the ambiguity. We divided the types into smaller components when the types covered multiple intents or merged if the types were redundant.

To make sure the taxonomy covered all information in how-to videos, we checked if any sentence contained information that could not be covered by the existing taxonomy. If so, we added additional types that encompassed the sentence and other similar content. After resolving conflicts and defining new information types, the new taxonomy would be used to reexamine the entire dataset.

Among the entire dataset of 120 videos, we started from an initial set of six videos and repeated the process until convergence was reached; (1) no new types were added and (2) no types were merged or

[2] HowTo100M dataset occasionally contains videos that are not exactly instructional, such as playing with toys or comparing two products.

split in the last iteration [130]. If these conditions were not met, we added six additional videos to the investigation. This resulted in an analysis of 48 videos to create the taxonomy.

4.2.2 Taxonomy

Through the iterative open coding, 21 **types** of information were identified. We further grouped the types into eight **categories** based on what function the types perform in a video. Below we explain the eight categories and the information types under each category in detail. For ease of reading, we denote the various hierarchies as follows: *Category*, and `Type`.

Greeting

Greeting category offers statements to start and end the video, such as hellos, channel introductions, Intro and Outro, with `Opening` and `Closing`, respectively. `Opening` includes beginning remarks and instructor/channel introductions, such as *"Welcome back to my channel!"* On the other hand, `Closing` gives parting remarks and wrap-up sentences, such as *"I hope you guys enjoyed this video, see you guys next time!"*

Overview

Overview category discusses the overall structure and information about the video. `Goal` is the main purpose of the video and its descriptions. For example, `Goal` of a cooking video may be, *"Today, we'll be making potato soup." Overview* also includes `Motivation`, which is the reasons or background information on why the video was created, such as *"Because everyone is getting a cold these days!"*. Finally, `Briefing` covers a quick rundown of how the goal will be achieved, such as *"I'll be doing a two-step process in this demonstration"*.

Method

Method provides core information required to complete the task. `Subgoal` outlines the objective of a subsection of the video, such as *"Now, let's prepare all our vegetables."*, without detailing specific directions that the user can follow. Rather, `Instruction` is the action that the instructor performs to complete the task that directly informs the user what they must do, such as *"Now, cut this rubber sleeve off."* `Tool` includes sentences that introduce or show the materials, ingredients, and equipment that will be used during the task, such as *"What we get usually is some cooking aluminum foil."*

Supplementary

Supplementary information suggests additional instructions or knowledge that aid the core instructions. `Tip` is information given to make the instructions easier, faster, or more efficient, such as *"This step is easiest to complete if you lower the headrest all the way down."* They are typically optional, but helpful advice that arises from the instructor's experience or knowledge. Meanwhile, `Warning` alerts the user on actions that should be avoided to prevent negative consequences, such as *"Don't get too wild with a hammer on there."*

Category	Type	Definition	Example from Dataset
Greeting	Opening	Starting remarks and introductions	*"Hey, what's up you guys, Chef here."*
	Closing	Parting remarks and wrap-up	*"Stay tuned, we'll catch you all later."*
Overview	Goal	Main purpose of the video and its descriptions	*"Today, I'll show you a special technique which is about image pressing."*
	Motivation	Reasons or background information on why the video was created	*"[...] Someone is making a very special valentine's day meal for another certain special someone."*
	Briefing	Rundown of how the goal will be achieved	*"I'm pretty sure that just taking a pencil and putting it over the front and then [...] that's going to do it."*
Method	Subgoal	Objective of a subsection	*"Now for the intricate layer that will give me the final webbing look."*
	Instruction	Actions that the instructor performs to complete the task	*"We're going to pour that into our silicone baking cups."*
	Tool	Introduction of the materials, ingredients, and equipment	*"I'm also going to use a pair of scissors, a glue stick, some fancy or regular tape."*
Supplementary	Tip	Additional instructions or information that makes instructions easier, faster, or more efficient	*"I find that it's easier to do just a couple of layers at a time instead of all four layers at a time."*
	Warning	Actions that should be avoided	*"I don't know but I would say avoid using bleach if you can."*
Explanation	Justification	Reasons why the instruction was performed	*"Because every time we wear our contact lenses, makeup and even dirt particles might harm our eyes directly."*
	Effect	Consequences of the instruction	*"And these will overhang a little to help hide the gap."*
Description	Status	Descriptions of the current state of the target object	*"Something sticky and dirty all through the back seat."*
	Context	Descriptions of the method or the setting	*"[...] The process of putting on a tip by hand [...] takes a lot of patience but it can be done if you're in a pinch."*
	Tool Specification	Descriptions of the tools and equipment	*"These are awesome beans, creamy texture, slightly nutty loaded with flavor."*
Conclusion	Outcome	Descriptions of the final results of the procedure	*"And now we have a dinosaur taggy blanket that wrinkles, so a fun gift for any baby on your gift giving list."*
	Reflection	Summary, evaluation, and suggestions for the future about the overall procedure	*"However, I am still concerned about how safe rubbing alcohol actually is to use so maybe next time, I will give vodka a try."*
Miscellaneous	Side Note	Personal stories, jokes, user engagement, and advertisements	*"Tristan is back from basketball. He made it on the team so it's pretty exciting."*
	Self-promotion	Promotion of the instructor of the channel (i.e. likes, subscription, notification, or donations)	*"So if you like this video, please give it a thumbs up and remember to subscribe."*
	Bridge	Meaningless phrases or expressions that connect different sections	*"And we're going to go ahead and get started."*
	Filler	Conventional filler words	*"Whoops."*

Table 4.1: Definition and examples of information types in our taxonomy. Minor errors from Speech-to-Text results in example sentences are corrected.

Explanation

Explanation elaborates on the reasons or consequences of the instruction to help users understand it more clearly. `Justification` is the reason why the instruction was performed. For example, the instructor may decide to use chicken breast because *"it has less fat than chicken thighs."* `Effect` refers to statements that explain the consequences of an action, such as *"Adding this activator will make the slime harden."*

Description

Description adds descriptions regarding the information relevant to the task, such as the state of the objects or the context of an action. `Status` describes the current state of the object or the target of the task. Sentences such as *"The car is making less noise."* is reporting on how the car is behaving currently and is thus `Status`. `Context` is the description of the method or the setting. For the method, the instructor may point out how arduous a task may be or explain how long it might take, such as *"It will take a while to come up."* For the setting, the instructor could mention, *"The room was really humid, so it took a while to dry."* Lastly, `Tool Specification` adds details and descriptions about the materials, ingredients, and equipment that may be mentioned in `Tool` or other parts of the video. The difference between the two types is that `Tool` merely establishes the usage of a tool (*"We'll be using some resin."*) while `Tool Specification` supplies other information or characteristics about the tool (*"This resin emits a lot of fumes."* or *"I'll leave a link of where I got it below."*).

Conclusion

Conclusion wraps up the video by showing the final outcome of the task and reflecting on the overall procedure. `Outcome` describes the final results of the procedure, such as *"Look how beautiful our cake turned out."* `Reflection` focuses on the summary, evaluation, and suggestions for the future. The following sentences, *"We made the batter, baked and iced it, and finally decorated it with some fruit."*, *"The process was so easy that even kids can do it."*, *"Next time, let's try using some honey instead of sugar."*, all fall under `Reflection`.

Miscellaneous

Miscellaneous refers to trivial information or phrases devoid of relevant information to the task. `Side Note` includes any sentences that mention personal stories, jokes, and advertisements or try to engage and communicate with the user, such as *"Comment down below what you think about this new look."* `Self-promotion` is the promotion of the instructor or the channel through the encouragement of likes, subscription, notification, or donation features common on creator-based video-streaming platforms, such as *"Please give it a thumbs up."* `Bridge` is meaningless phrases or expressions that connect different sections or phrases, such as *"Let's move onto the next part."* Finally, `Filler` is the conventional filler words prevalent in spoken language, such as *"um"*, *"uh"*, or *"well."*

4.3 Dataset

To validate the taxonomy, we applied the taxonomy to the remaining 72 videos and contribute the type-labeled 120 videos as a dataset. The dataset can be used to model automatic type detection pipelines or be leveraged to explore various system design opportunities that apply our taxonomy. This section describes the dataset and the following section describes the analysis we performed on the dataset to investigate how videos are structured.

4.3.1 Method

We applied the taxonomy to the remaining 72 videos (5.9k sentences) to validate the taxonomy and contribute a dataset. Two external fluent English-speaking annotators coded 72 videos based on the taxonomy (6 videos each from 12 genres), where they independently coded the sentences with their types and merged the labels into agreed-upon final labels. Similar to the taxonomy construction process, the annotators watched the videos while labeling the type of each sentence to understand the context behind each sentence and to clarify any errors in the transcript. The annotators were asked to split the sentence if they thought it contained more than one information type. The total number of split sentences was around 1% of all sentences. The two annotators and one of the authors met regularly to discuss ambiguous cases and resolve conflicts. For the last 42 videos (3.4k sentences, with the remaining videos used for training), the two annotators had Cohen's Kappa score of 0.78, which shows a satisfactory level of agreement [4]. After the score was calculated, conflicts were resolved by a discussion between the two annotators and one of the authors. The coding process took approximately 70 hours per coder.

4.3.2 Dataset: HTM-Type

We release a dataset, HTM-Type[3], which contains a total of 9,918 type-labeled sentences (mean=82.65, SD=21.8) from 120 videos selected from the HowTo100M dataset [120]. It consists of 10 videos from each of the 12 genres identified by HowTo100M. All videos are longer than 5 minutes and published within the last five years (2017 and onward). The average length of the videos is 7 minutes 3 seconds (SD=1 min 35 sec, min=5 min 1 sec, max=14 min 49 sec), totaling 14.1 hours. The average portion of spoken language is 82.4%, representing the average portion of the entire video in which the author talks (min=50.5%, max=97.6%). The dataset denotes for each sentence the id, publication date, duration, and genre of its video, as well as start and end time stamps, and type and category categorization.

4.4 Analysis

To understand the structure of how-to videos, we analyzed the HTM-Type dataset in three different aspects: **(1)** how each information type is distributed across the dataset, **(2)** how the video style affects the type distribution, and **(3)** how information type distribution relates to time.

4.4.1 Method

For all three analyses, we first identified the proportion of each information type in a video by calculating the start and end timestamps of each labeled sentence. Afterward, we divided the time portion of each type by the total time of the video containing narration to obtain the final proportion.

(1) The first analysis aims to observe how the information types are distributed throughout the how-to videos. We calculated the average distribution of each type across the entire dataset by dividing the total time proportion of each type by the number of videos.

[3] Abbreviated from HowTo100M-Type

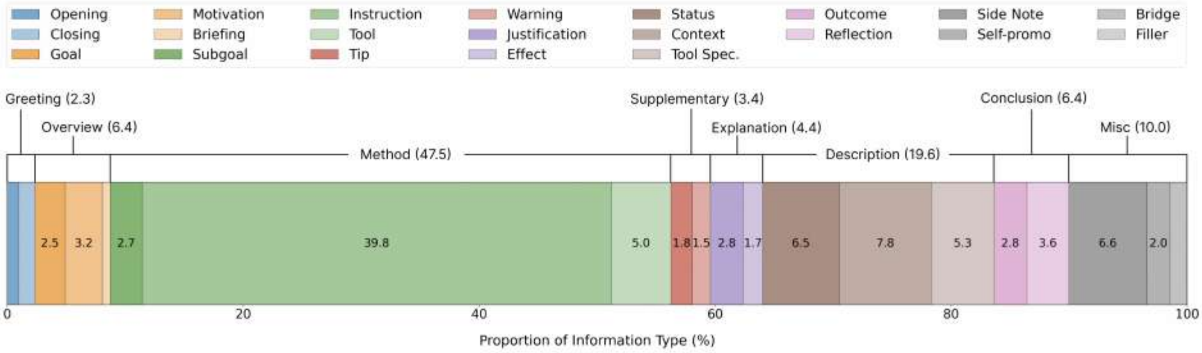


Figure 4.1: Distribution of Categories and Types of all videos in HTM-Type. Categories are denoted above the types using group brackets. Only proportions greater than 1.5% are written in text. `Instruction` makes up 39.8% of the total video, suggesting that the majority of the video contains information that does not directly give actions for the user to follow. The results illustrate the large diversity of information types in how-to videos.

(2) The second analysis examines how the video characteristics affect the information distribution along two different attributes: task type and narration style. We chose task type and narration style specifically as the analysis axes as they require different strategies by the instructor in providing the information. For example, explaining how to fix a car likely attributes a larger portion of the video to describing the situation in comparison to baking cookies.

To compare whether video characteristics affect the distribution of the information type, we performed the Kruskal-Wallis test for each of the two attributes with an α value of 0.05 for each category. We further performed the Kruskal-Wallis test on types within the different categories if the category showed a significant difference. To confirm which specific video characteristics differed from one another, we further performed post-hoc Dunn's test with Bonferroni adjustment on significantly different categories or types.

(3) The third analysis aims to investigate any specific patterns that may appear in the temporal distribution of each category. To do so, we normalized video time to [0, 1000] seconds to align all the videos in the dataset. Then, we counted each type occurrence across all 120 videos for every second on the normalized timeline. As none of the videos in the dataset are longer than 1000 seconds, the normalization will not drop any labels. Afterward, we calculated the range on the normalized timeline that contains data points between the 5th and the 95th quantile for category.

4.4.2 Results

Information Distribution in How-To Videos

We first investigated the composition of the dataset to look into how the diverse information is distributed over how-to videos. The results for categories and types are shown in Figure 4.1. The average number of types in a video is 7.25 for category and 14.57 for type, signifying that the videos comprise a wide variety of information. Additionally, the large variance of the types suggests diverse variations in how the information is composed within instructional videos.

On average, the results show that almost half of the video comprises *Method* (47.5%, SD=16.9%). Looking at the type level, `Instruction` makes up 39.8% of the total video, meaning that the majority of the video contains information that does not directly give actions for the user to follow. The ratio shows a resemblance to the percentage of visually alignable narration as explained by Han et al. [65] (30%), which is a narration that is visually demonstrated or shown in the video. As instruction usually entails the majority of the visual information, the similarity may imply some correlation.

Information Distribution Based on Video Characteristics

We then analyzed how the video characteristics (i.e. task type and narration style) affect the information distribution. Through the analysis, we found that the composition of information types in a video differed by its characteristics, which we describe below.

Task Type The first aspect examined is the type of task completed. Through an iterative process, we found three different task types: Creating, Fixing, and Using. Creating refers to tasks whose primary goal is to craft or make a final product, such as cooking or woodworking. Fixing tasks address a problem and improve the state of an object or a situation. Using tasks aim to demonstrate how a tool or equipment is supposed to be used. Our dataset contains 82 videos for Creating, 27 videos for Fixing, and 11 videos for Using.

The results of the Kruskal-Wallis test show significant differences between the tasks for *Description* (H(3)=21.696, p¡0.001) and *Miscellaneous* (H(3)=10.435, p=0.015). Further performing the Kruskal-Wallis test on the types in the *Description* and *Miscellaneous* categories reveals that `Status`, `Context`, and `Side Note` are significantly different.

Further performing post-hoc Dunn's test with Bonferroni adjustment showed that Creating-Fixing and Using-Fixing pairs for `Status` and Creating-Fixing for `Context` are significantly distinct in their distributions ((Z=-2.680, p=0.022), (Z=3.126, p=0.005), and (Z=-2.443, p=0.043) respectively). Fixing (10.0%) has a greater proportion of `Status` than Creating (5.7%) and Using (3.3%). For `Context`, Fixing (11.6%) is greater than Creating (6.2%) by 5.4%. Such differences can be explained by the tendency for Fixing tasks to require more descriptions of the target object. Conveying `Status` in Fixing videos lays the necessary foundation to communicate the instructions effectively. Likewise, Fixing has more explanations than Creating about the method and the setting because the user needs to fully grasp the current circumstances before they can improve upon them.

Narration Style The second aspect is the narration style of the video. Videos were classified by how the instructor provided verbal information — whether the narration was spoken in real-time with the action or dubbed afterward. We found 78 videos are real-time narrated and 42 are dubbed videos.

The results of the Kruskal-Wallis test on the categories showed that *Method* and *Description* show significant differences between the narration styles ((H(1)=6.602, p=0.01) and (H(1)=7.036, p=0.008), respectively). To figure out how each type distribution differs within the two categories (*Method*, *Description*), we further performed the Kruskal-Wallis test for each type in the categories. `Instruction` and `Tool specification` have significant differences in their distributions ((H(1)=7.568, p=0.006) and (H(1)=4.043, p=0.04), respectively). When comparing the absolute value of each type proportion on average, for `Instruction`, dubbed videos (45.0%) contain an 8.1% greater portion than real-time narration videos (36.9%). On the other hand, for `Tool Specification`, real-time narration videos (5.9%) have more than dubbed videos (4.2%).

The differences show that video styles can affect the distribution of information. Real-time narrated videos contain a larger portion of descriptions such as `Tool Specification`, `Status`, and `Context`. One

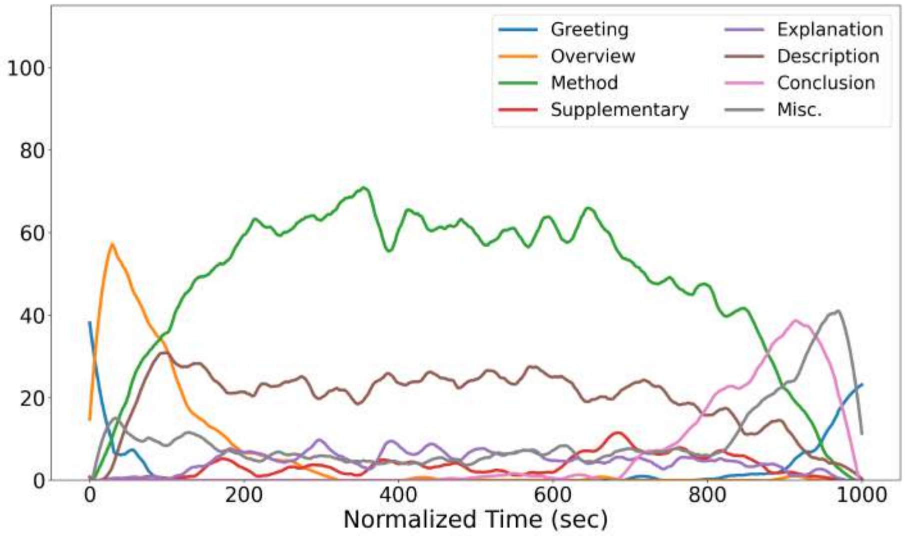


Figure 4.2: The number of labels for the category along normalized time. *Greeting*, *Overview*, *Conclusion*, and *Miscellaneous* show clear positional preferences while *Method*, *Supplementary*, *Explanation* and *Description* are widely distributed.

possible reason may be that the instructor dedicates more time to explaining the current status quo as they actually perform the task.

Information Distribution Based on Time

We then analyzed the temporal distribution of each category to see if they showed any specific patterns. We visualized the data with a time-series graph (Figure 4.2).

The results show that certain categories have a positional preference. *Greeting* shows skewed distributions towards both ends of the video. Such a trend reflects the tendency for instructors to begin or end their videos by greeting their audiences. *Overview* occupies the first (23.8%) of the video, as it covers the overall structure or encompassing details of the video. Meanwhile, *Conclusion* lies in the last (28.0%) of the video. In contrast, *Method* (11.1% to 85.3%), *Supplementary* (16.9% to 86.3%), *Explanation* (16.8% to 87.2%) and *Description* (8.5% to 86.9%) are relatively evenly distributed towards the middle of the video. Finally, *Miscellaneous* extends throughout the video (4.8% to 98.0%) with a noticeable increase at the end (Figure 4.2), attributed to the abundance of self-promotion and side notes (e.g., outtakes).

4.5 Taxonomy as Analytical Framework

In this section, we demonstrate how our taxonomy can serve as a conceptual and analytical framework for understanding existing systems that support video navigation. Existing video navigation systems are designed to address specific user needs. Our taxonomy provides an opportunity to analyze the information types that each system focuses on. Such an analysis can be used to identify important information types that best fit the users' context and also reveal information types that are underexplored by existing systems.

For instance, ToolScape [84] and MixT [34] have identified step-by-step information (`Subgoal`) with representative images for each step (`Status`) to allow users to navigate videos based on important milestones. To better support navigation in a specific video genre, VideoWhiz [127] has extracted ingredients (Tool) and intermediate outcomes (Status) in food recipe videos, and Truong et al. [167] has leveraged makeup tools (Tool) in makeup tutorial videos. To support users navigating videos in a setting where they use voice commands, RubySlippers [24] has allowed users to refer to objects (Tool) and actions (Instruction) that appear in the video.

As such, existing systems have leveraged different information types to address specific needs in video navigation, which we list more in Table 4.2. We can see that the types in the *Method* category (i.e. Subgoal, Instruction, and Tool) are commonly used, while Goal, Status and Outcome are also used to some extent. At the same time, our investigation reveals that the other information types are underexplored by existing systems, such as Motivation or Context. We believe that future systems can establish important units based on the identified information types catered to user needs.

System	Type	Explanation
ToolScape [84], MixT [34], Fraser et al. [56]	Subgoal, Status	Presenting step-by-step information (Subgoal) with representative images for each step (Status)
Truong et al. [167]	Tool, Instruction, other types	Labeling segments as tool introductions (Tool), makeup application (Instruction), or commentary (other types)
VideoWhiz [127]	Tool, Subgoal, Status, Outcome	Presenting ingredients and equipment used in a recipe (Tool), visual milestones (Status, Subgoal), and the appearance of the final output (Outcome)
RubySlippers [24]	Tool, Instruction	Allowing users to refer to objects (Tool) and actions (Instruction) that appear in the video
Pause-and-Play [148], SoftVideo [191]	Instruction	Segmenting software tutorial videos into actionable steps (Instruction)
Weir et al. [175]	Goal, Subgoal, Instruction	A breakdown of a task into the goal (Goal), subgoals (Subgoal), and individual steps (Instruction)
Yang et al. [187]	Tool, Instruction	Segmenting recipe videos into actions (Instruction) and visualizing their dependencies as well as ingredients used (Tool) throughout the video.

Table 4.2: Example systems that support video navigation and information types associated with each system.

4.6 Exploratory User Study

From the preliminary analysis presented in Section 4.5, we demonstrate how our taxonomy could serve as an analytical framework for understanding existing video navigation systems. To further explore the potential of the taxonomy, we conducted an exploratory user study. Our study aimed to investigate how users would leverage the information types for navigating videos, by exposing information types to users and allowing them to navigate videos using the information types as a control mechanism. Through the study, we demonstrate the usefulness of the taxonomy both in accessing desired content and as a tool for observing and analyzing users' navigational behavior. We chose not to conduct a comparative study because the purpose was not to evaluate the video interface itself but rather to highlight the potential of the taxonomy in supporting video navigation, an aspect that has been underexplored in previous research. Below we explain the research probe used in the study, the study procedure, and the results.

4.6.1 Research Probe

As the apparatus of the study, we built a video interface that supports navigation based on information types (Figure 4.3). Users can see the video on the left (Figure 4.3a) and transcripts of the video on the right (Figure 4.3b). In the transcript panel, users can see each sentence of the transcript along with its timestamp and information type. The type label is color-coded based on the category of the taxonomy. The timeline also shows the same information below the video (Figure 4.3c). Each segment is color-coded based on its category and users can hover over each segment to see its type (Figure 4.3d). The type of the current segment is always shown right next to the progress bar. Users can click either on the timeline or the script to navigate through the video. Finally, users can filter segments based on their type or category in the Filter panel (Figure 4.3e). Here, we grouped the categories into four high-level sections to help users better organize the types and categories: Intro, Procedure, Closing, and Miscellaneous[4]. We organized the categories based on their temporal positions reflecting our analysis in Section 4.4.2. Once users select certain types from the Filter panel, only the filtered segments are shown in the transcript panel and in the timeline. The video player automatically skips unselected portions.

4.6.2 Study Procedure

We recruited nine participants (6 male, 3 female, mean age=24.1, SD=2.26, min=22, max=29) through an online recruitment posting. All the participants watch how-to videos regularly, at least once a week. Participants performed three types of tasks: Search, Summarize, and Follow. These tasks represent real video-watching scenarios and are commonly used in evaluating video navigation systems [24, 82, 167, 84]. We chose three videos from HTM-Type that cover different tasks: Cooking[5], Slime[6], and Illustrator[7]. The Cooking video teaches how to make soft-boiled eggs. The Slime video explains how to make cloud slime. The Illustrator video demonstrates how to convert raster images to vector images. To minimize learning effects, different videos were used in each task. The videos used for each task were counterbalanced between the participants.

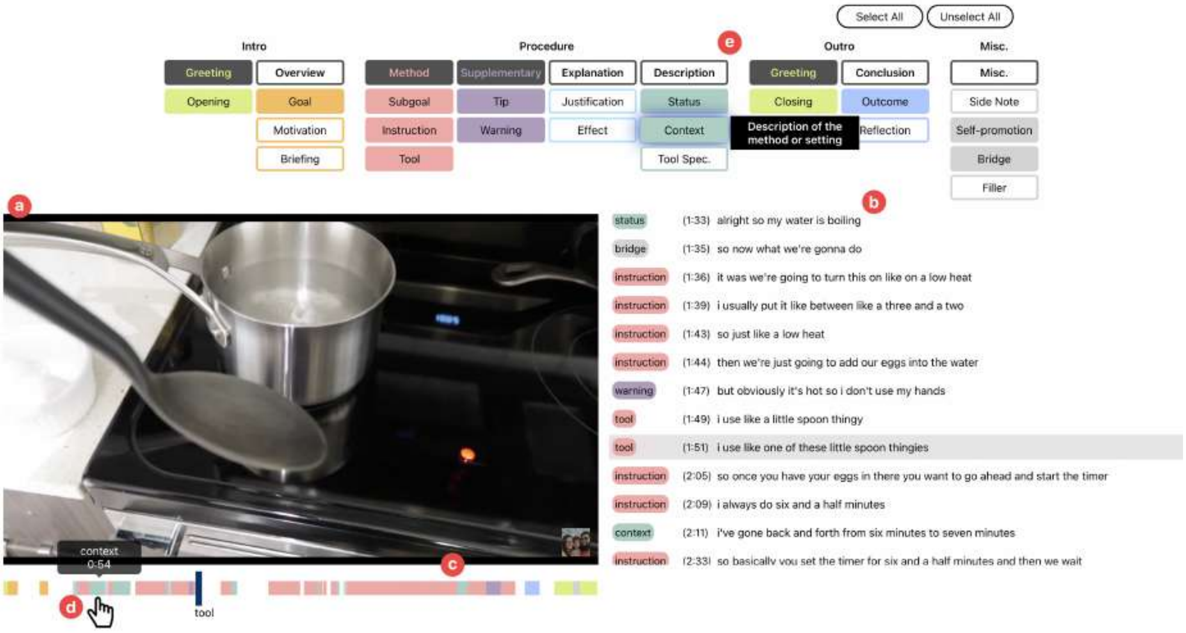
- **Search** task asked participants to find an answer to a given question from the video. For example, for the Illustrator video, the task asked: "*To make the image more cartoonish, which feature do you need to adjust?*" There were three search questions for a video.

[4] In the process of grouping, `Opening` and `Closing`, which belong to the *Greeting* category, were divided into Intro and Outro, respectively.

[5] youtu.be/6CJryveLzvI

[6] youtu.be/Rcsy2HRuiyA

[7] youtu.be/_Yb6xLqvsf0



status (1:33) alright so my water is boiling
bridge (1:35) so now what we're gonna do
instruction (1:36) it was we're going to turn this on like on a low heat
instruction (1:39) i usually put it like between like a three and a two
instruction (1:43) so just like a low heat
instruction (1:44) then we're just going to add our eggs into the water
warning (1:47) but obviously it's hot so i don't use my hands
tool (1:49) i use like a little spoon thingy
tool (1:51) i use like one of these little spoon thingies
instruction (2:05) so once you have your eggs in there you want to go ahead and start the timer
instruction (2:09) i always do six and a half minutes
context (2:11) i've gone back and forth from six minutes to seven minutes
instruction (2:33) so basically you set the timer for six and a half minutes and then we wait

Figure 4.3: Our research probe used in the user study. (a) Users can see the video. (b) Each sentence of the script is shown with its timestamp and information type. Each type label is color-coded based on the category. (c) The same information is shown in the timeline. (d) When users hover over each segment, they can see the type and (e) its definition in the Filter panel. Users can filter segments based on their type or category in the Filter panel. Only the filtered segments are shown in the transcript panel and the timeline.

- **Summarize** task asked participants to summarize the main points of the video while skimming through it. We asked participants to assume that they are making written instructions from the video content. We gave participants freedom in the content and format of the summary.
- **Follow** task asked participants to follow the task in the video. We prepared the tools used in each video. For the cooking video, we simulated the cooking environment with hand-made apparatus such as a stove made of paper.

We first gave a tutorial on the system to the participants. After explaining its features, participants tried out the system with a video that was not used in the three tasks. Then, we explained the taxonomy presented in the system. After explaining the definitions and examples of each type, participants watched a video with our interface from beginning to end to get used to the taxonomy. Participants were subsequently asked to perform three tasks in the following order: Search, Summarize, and Follow. To accurately evaluate the role of information types in each task, participants were not allowed to use the browser's native search function (i.e., Ctrl+F) in the transcript. After each task, we asked a few questions about their task strategy. After all the tasks were done, we conducted a semi-structured interview and survey, asking about their experience and perceptions of the taxonomy. Participants were compensated with 20,000 KRW (∼15 USD) for a 1.5-hour-long study.

4.7 Results

The participants were able to find and use appropriate types or categories of the taxonomy to complete the tasks. Below we explain how they used the taxonomy and the information types they perceived as important in detail. Then, we discuss how the participants perceive the prototype and the taxonomy.

4.7.1 How Taxonomy Was Used in Each Task

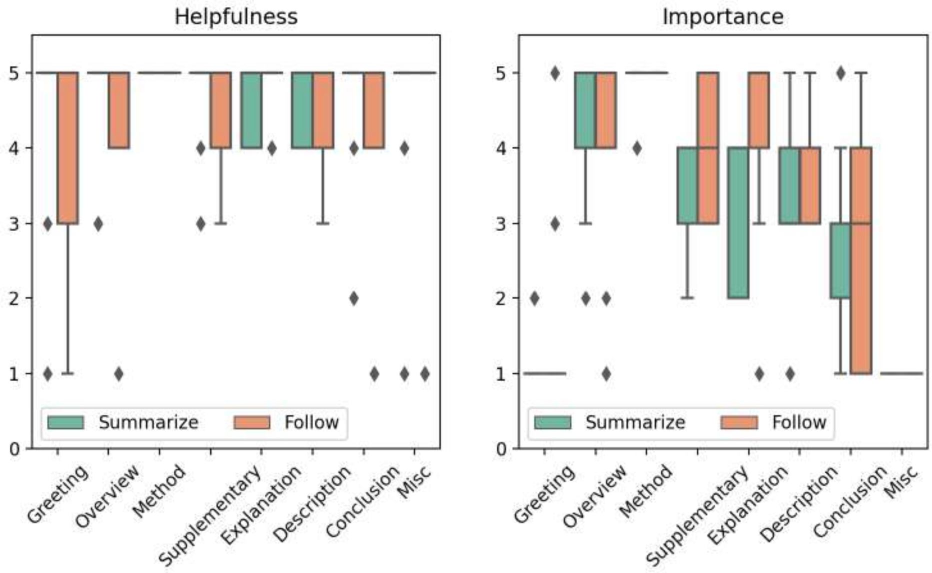


Figure 4.4: Helpfulness (left) and Importance score (right) of each category in the Summarize and Follow task.

Search

The participants' strategy to search for the answer to questions was to relate a given question to a type and filter the video according to the type. For example, for a question asking about how the recipe is different from others (Slime), P3 thought it would be described when the instructor talked about the goal. Thus, he filtered the video to only see `Goal` and found the answer. For this task, participants looked for different information types depending on what each question asked. All the participants were able to match at least two questions out of three correctly to corresponding types (mean=2.44/3, SD=0.53), and thus found answers effectively.

Summarize

The participants actively used the information types and found them helpful when summarizing videos. In response to 5-point Likert scale questions about how helpful each category and type's existence was (including the removal of them), participants indicated that the existence of all of the categories (mean=4.61/5) and types (mean=4.66/5) were useful, when asked about each category and type individually (Figure 4.4-left).

When asked about the importance of each category in summarizing videos, they rated *Method* and *Overview* as the top two categories that contain the most important information (Figure 4.4-right, 4.89 and 4.11/5, respectively). Not surprisingly, all the participants looked for the Method category, as they are the main points of videos. Regarding *Overview*, P3 said, *"I looked for Overview because I felt it is necessary to include the purpose of the task when summarizing the video content."*

From the per-type evaluation, the participants rated `Instruction`, `Subgoal`, `Tool`, and `Goal` as the top four important types (4.89, 4.78, 4.78, and 3.89/5, respectively). Regarding `Instruction`, all the participants included instructions in their summaries (n=9) as they are the essential information in how-to videos. Interestingly, participants not only used the `Subgoal` information to organize their summary by subgoal unit (P7) but also to check and see if they have missed anything at the end (P3, P4). Participants also included the tools used in the video (n=5) and the goal of a video (n=6) in their summaries, along with a description of the goal (n=2) and warning (n=1). Additionally, some participants (P1, P6) looked for `Reflection`, expecting the part to provide a summary, although the video did not include any summary information and thus rated low (2.67/5). All the types under the *Greeting* and *Miscellaneous* categories are rated the lowest (mean=1.61/5), as they do not include any task-relevant information.

Follow

In following the task performed in the videos, the participants perceived the information types to be helpful. In response to 5-point Likert scale questions about how helpful each category and type's existence was (including the removal of them), participants indicated that the existence of all of the categories (mean=4.35/5) and types (mean=4.32/5) were useful, when asked about each category and type individually (Figure 4.4-left).

When asked about the importance of each category in following the videos, they rated *Method*, *Supplementary*, and *Explanation* to be the top categories that contain important information (Figure 4.4-right, 5, 4.11, 4.11/5, respectively). Not surprisingly, participants thought *Method* contained most of the information they should follow. After *Method*, the participants perceived *Supplementary* and *Explanation* to be important, which was different from the Summarize task. The participants thought the *Supplementary* category which includes `Tips` and `Warnings` to be important. P4 said, *"I thought tips and warnings are too detailed information for the Summarize task. However, they were necessary when following the video as they might contain important notes."* They also found the *Explanation* category which includes `Justification` and `Effect` to be helpful. P3 said, *"It was helpful to know the reasons behind instructions because then I can apply instructions to my context adaptively. For example, if I understand that the reason instructor boils eggs for six minutes is that it's the medium part of being too runny and firm, I can adjust the duration according to my taste."*

From the per-type evaluation, participants rated `Instruction`, `Subgoal`, and `Tool` as the top three important types (4.89, 4.78, and 4.45/5, respectively), followed by `Effect`, `Tip`, `Warning`, and `Justification`, and `Status` (4.11, 3.89, 3.89, 3.67, and 3.67/5, respectively). The participants used `Effect` and `Status` to make sure they are following correctly. P7 said, *"I considered Effect to be important because I wanted to check that the consequences of an action explained in the video are actually shown in my context."* Similarly, P8 said, *"I looked for Status to see if there is a desired state, and if so, I would have liked to refer to it when following."* We could see that the participants mainly focused on instructions while looking for additional information when following videos.

4.7.2 Effect of Taxonomy on Video-watching Experience

All the participants appreciated that the system enabled selective watching of videos. P8 said, *"When watching how-to videos, I usually watch the video at twice speed or skip parts because there is a lot of unrelated information. It was nice to be able to get rid of useless information."* Selective watching can also be helpful in repeated watches. P5 said, *"I think the system will be helpful especially when you watch a video again and again. For complex tasks like repairing, it is hard to perform the task at once. If you know where to watch repeatedly, it will be efficient."*

Some participants compared the selective watching feature to YouTube's Chapter where it segments a video into meaningful sections [194]. P2 and P4 appreciated that our system offers more details. P2 said, *"In YouTube, we can also skip some parts but it's based on topics. We still have to search within a topic by trial and error, to see the exact part I want."* However, other participants mentioned that the amount of higher-level information they could perceive for each section was limiting. P5 said, *"I could skip parts with the prototype, but YouTube chapters indicate subgoals better with a concise title, which makes it easier to access desired parts."*

The information type was helpful in grasping the overall content. P6 said, *"By looking at the timeline, I was able to quickly understand how the whole video is composed of. For example, from the timeline, I was able to figure out the style of the video, such as whether this video has a lot of intro or outro, or whether it has a lot of unrelated miscellaneous information."* It also allowed the participants to grasp the main points quickly. P8 said, *"I was able to understand the flow of the video quickly, by looking at the instructions only."* Participants also thought that it highlights important information for them. P5 said, *"Warnings are important information but they can be unnoticed easily. The prototype helped me identify them."*

4.7.3 Perception Toward Taxonomy

Overall, the participants were able to understand the meaning of each category and type well (Category mean=4.86, Type mean=4.75). They mentioned that the types were intuitive (P3), and they were able to see the reasoning behind the categorization (P9). All the participants mentioned that each sentence was well-matched with appropriate types, except for a few that were subjective. One feedback that many participants had in common was that the categories would be enough for filtering the video content (P1, P3, P4, P9). While types allowed for more precise control (P6), it was burdensome to recall the meaning of each type and click them one by one due to the large number of types (P9). In the same context, several participants also suggested indicating whether a type exists in the video so that they do not have to manually click to see if it is in the video. As such, when designing systems that display taxonomic information, we need to consider ways to reduce users' cognitive burden.

4.8 Discussion

In this paper, we present a taxonomy of information types in how-to videos. We first demonstrated how our taxonomy can serve as an analytical framework for existing video navigation systems. We then investigated the utility of the taxonomy in video navigation through a user study. In this section, we first reflect on the user study and discuss findings. We then discuss how the taxonomy enables various video-related tasks and support the learning experience, and suggest opportunities for future work.

Information Type That Fits the User's Needs

While the essence of how-to videos is information that explains how to perform a step (i.e. `Instruction`), our taxonomy identifies a total of 21 information types that span instructions and beyond. From our user study, we could see that the participants used different information types for each task. In the Search task, they were able to actively match the corresponding information types to each question, finding answers effectively. In the Summarize task, *Method* and *Overview* were considered important – the participants used *Overview* to summarize the goal and overall approach. In the Follow task, in addition to *Method* that provides core information required to complete the task, the participants also considered *Supplementary* and *Explanation* important in getting additional information needed in following the video.

Just as important types vary depending on the task, our study also suggested that meaningful information types can depend on various factors such as the topic of the video or the user's level of expertise. P6 said, *"In videos teaching how to play tennis, justification or effect might be more important than just instructions. It is important to understand WHY a certain movement is needed to actually understand and follow the movement."* It also echoes Semeraro et al.'s finding on instructional videos for physical training, where having verbal cues helped users contextualize the movement [153]. Users' familiarity with the topic also affects which information types they focus on. For example, P8 was unfamiliar with Adobe Illustrator so she checked *Overview* for goal descriptions when following the video. She said, *"I would have skipped the part if I were familiar with the program."* Future work will need to investigate relevant information types depending on the topic and user context.

Moreover, some participants suggested further specification of instructions based on their importance. In how-to videos, there are optional or conditional instructions that users can choose to follow or not according to their preferences or environment. P6 mentioned that *"I thought all the instructions are necessary, but there were some instructions that I didn't need to follow. It would have been nice if it had been marked."* In fact, four participants additionally marked optional or conditional instructions in their summary when performing the Summarize task, which implies the importance of such information. As such, future work can specify the instruction types to support users' detailed needs.

In summary, our findings suggest that 1) information types other than *Method* can also play an important role in accessing desired information, which opens up opportunities for future systems to take into account a variety of information types. Our findings also suggest that 2) relevant information types can be different depending on the task, topic, and user context, which future work can investigate more in depth to support users' different needs. We hope that our taxonomy can serve as a starting point for such investigations.

Applications of Taxonomy in Video Tasks

The taxonomy can accelerate the design process of multiple applications if videos were labeled by information types. We examine possible applications in three of the most commonly performed video-related tasks: Authoring, Viewing, and Analysis. The creator first produces a video (Authoring), and then viewers watch it (Viewing). The creator can analyze the video content or viewership to improve the original video and make decisions about upcoming content (Analysis). We discuss how our taxonomy enables various applications in each of these tasks.

Application		Explanation	Example
Authoring	Editing	Removing or fast-forwarding parts of the video	Cut out irrelevant parts of the video (`Side Note`)
	Annotation	Adding visual effects or captions to the video	Highlight important parts of the video (`Tip`, `Warning`)
Viewing	Navigation	Supporting users to find relevant portions of the video	Repeat an instruction segment or jump to the next instruction (`Instruction`)
	Summarization	Providing a summary of the main points of the video	See an outline of how the goal is achieved (`Subgoal`, `Instruction`)
	Search and Selection	Supporting users to make a decision on which video to watch	See if one has required tools to follow the video (`Tool`)
Analysis	Feedback	Providing feedback to the author of the video about the content	Inform the author about how structured the video is (`Subgoal`)
	Comparison	Comparing content between multiple videos	Compare how approaches toward a same goal are different (`Instruction`)

Table 4.3: Possible applications of the taxonomy in video authoring, viewing, and analysis.

Authoring

Having a video labeled by the taxonomy can foster the video editing process. For example, instructors can find fillers or side notes that they have made, thus removing or fast-forwarding the parts if necessary. They can also add visual effects to parts that need extra attention, such as tips or warnings, or make transition effects when moving to the next step introduced by a subgoal. They can also add subtitles or textual descriptions and style them differently, depending on what and how much they want to emphasize [100].

Our taxonomy also aligns with the components that facilitate video editing found in previous papers. DemoCut [35], a video editing system designed for how-to videos of physical demonstrations, supports five types of markers to assist in video editing: Step, Action, Closeup, Supply, and Cut-out. The system segments a video and applies editing effects based on the markers. Our taxonomy aligns with several types of the markers, such as Step (`Subgoal`), Action (`Instruction`), Supply (`Tool`), or Cut-out (`Miscellaneous`).

Viewing

Our study revealed that the taxonomy can improve users' viewing experiences by enabling them to quickly find and skip irrelevant information based on the category and the type. Our findings echoes with Chang et al.'s finding on the types of jumping in how-to videos: Reference Jump (reminding users of past content), Replay Jump (re-watching a segment of the video), Skip Jump (skipping less interesting content), and Peek Jump (skipping ahead to see what to expect) [26]. Reference and Replay Jumps can happen around `Instruction`, to clarify any confusion and better understand the instruction. Skip Jump can happen around `Greeting` or `Side Note`, where a user wants to skip task-irrelevant parts. Lastly, Peek

Jump can happen around `Status` or `Outcome`, where a user wants to see intermediate or final outcomes.

Our taxonomy can further support video navigation by segmenting a video into meaningful sections, by leveraging `Subgoal`, `Status`, or `Bridge` information. P9 said, *"If we have the Goal and Subgoal information, I think the video can be divided by each section like a table of contents. I would have liked it."* P8 mentioned the possibility of using `Status`. She said, *"If Subgoal remarks the start of a step, I thought Status remarks the end of a step. It showed intermediate outcomes."* One can also leverage `Bridge` as it may signal transition to next chapter. As such, we can leverage meaningful information types to make navigation easier.

The taxonomy can also be useful when summarizing a video. As observed from our user study, users could choose the relevant information such as `Goal`, `Tool`, or `Instruction` to summarize the main points. They can also see a succinct summary explained by the author with `Briefing` or `Reflection` or an outline of how the goal is achieved with `Subgoal`. We can also make the summary generation process interactive by allowing the users to choose the information type that they want to see in a summary. In this way, we can give users more control over the summarization process beyond the time budget [76].

Lastly, our taxonomy can help users make an informed decision when selecting videos to watch. Users can use certain information types to assist their decision. P3 said, *"I would check Overview, Tool, and Conclusion first when deciding on whether to watch the video or not. I would check Overview and Conclusion to see if I like the method and outcome, and I would check Tool to see if I have all the required tools."* They can also see the proportion of information types to make a decision. P8 said, *"I don't really like videos that have a lot of irrelevant information. I would filter out videos that have a high portion of Miscellaneous information."* The taxonomy can also be used to recommend videos, providing explanations of recommendations such as conciseness or required tools. As in Inel et al.'s work which provides explanations of a video summary [73], it will help users understand the video with transparency.

Different users can rely on different information types based on their navigational or learning needs. With our taxonomy, we believe that users will have more control and agency in navigating, summarizing, and selecting videos with more informed decisions.

Analysis

Our taxonomy can provide a systematic way to help instructors reflect on their videos by analyzing content, viewership, and watching patterns. Receiving feedback on a video is key for authors in improving their videos [140]. Researchers have proposed several systems for providing feedback on videos, such as a script-based review system [140] or a system that analyzes accessible factors of a video [143, 109]. By applying our taxonomy to their videos, the author can see how focused the video is (e.g., Do I have too many `Side Notes`?) or how structured the video is (e.g., Do I mention enough `Subgoals`?). It can also give feedback on its accessibility, by looking at how descriptive the video is (e.g., Are there an adequate number of `Descriptions`?) [109]. Authors can also see which information type received more attention from viewers, and make informed decisions about the content revision and production.

The taxonomy can also enable comparison between multiple videos. With an increasing number of videos, many systems have been proposed to enable the exploration and analysis of large collections of videos [114, 59, 42]. However, one of the challenges in comparing videos is the complexity of the size and items to be compared. Tharatipyakul et al. proposed video abstraction as a way to reduce such complexity [163]. Our taxonomy enables abstracting a video such as by taking `Instructions`, thereby enabling efficient comparison between videos. It will allow identifying commonalities and differences in approaches toward the same goal [23, 25] or classify workflows at scale [170].

Supporting the Learning Experience

Understanding the information types in videos can help users in organizing the information. Mayer's multimedia learning theory suggests that learning material should have an understandable structure and guide the learner in making a mental model (Active processing principle) [115]. He suggests that it is helpful to know how information models can be structured. We believe that our taxonomy can contribute to structuring information in videos by organizing the information based on their kind, and thereby help the learning process of users.

Our taxonomy also includes information types that are critical to effective instructional content. According to Morain and Swarts [123], successful tutorial videos begin with an overview of what is to be accomplished (`Goal`, `Briefing`), explain what is accomplished (`Subgoal`) and reasons for performing a step (`Justification`), and describe details such as the tool selection (`Tool`), the settings (`Context`), and the outcomes (`Outcome`). Identifying meaningful information types for learners can ultimately extend their learning experiences beyond following along.

Furthermore, our taxonomy shares several components with the taxonomy of information types in lecture videos. Although how-to videos and lecture videos differ in the type of knowledge they convey (e.g. procedural vs. declarative), they share the commonality of conveying instructional information. Comparing our taxonomy to Espino's investigation on the taxonomy of verbal information in MOOC videos, there are several common components: 'Opening/closing shot' (`Opening`, `Closing`), 'Overview of the contents' (`Briefing`), 'announce following section' (`Subgoal`), and 'Justify/motivate content' (`Justification`, `Motivation`) [50]. We can see that our taxonomy identifies major components that aid learners in their learning process.

Technical Pipeline

To foster leveraging our taxonomy and developing applications discussed in Section 4.8, it is essential to develop a technical pipeline that classifies segments of a video into the information types of the taxonomy. As one of the approaches, we can leverage the few-shot learning technique on transcripts of a video with large language models such as GPT-3 [19]. However, since our taxonomy is not only based on verbal information but verbal information that considers visual information, multimodal learning that takes visual information into account might yield better accuracy. The hierarchy of our taxonomy (Category and Type) enables Hierarchical Classification as well. We hope our dataset containing 9.9k sentences labeled according to the taxonomy can be served as a useful starting point to build such technical pipelines.

Limitations and Future Work

In our study, we chose verbal utterances as a primary source of information. This is because how-to videos usually have content creators explaining verbally how to perform a task [35], with an explicit intention of explaining the visual content [120]. They also give additional information that is difficult to be delivered visually. Due to the unique and extensive role of verbal information in how-to videos, we presumed that it would cover a wide range of information and thus chose it as our scope.

However, videos are multimodal and visual information also plays an important role [123]. Although we considered visual information when annotating each sentence to understand context, it does not cover information types that only visuals can convey. For example, visual information can describe instructions in more detail, sometimes accompanied with annotations that describe emphasis on objects or provide more detailed information of a tool used [35]. It would be interesting to investigate videos that deliver information only through a visual channel to understand the capacity of information types that visuals convey. Furthermore, verbal and visual information might not always align with each other [65, 35]. For example, an instructor can verbally share instructions first and then visually demonstrate them later. As such, future work can incorporate visual information in how-to videos for a more comprehensive taxonomy and analysis.

Also, while our taxonomy is based on diverse videos in terms of topics, styles, and production methods, they were YouTube videos whose lengths are between 5 minutes and 15 minutes. It may be that some types in the taxonomy are specific to YouTube videos (e.g., `Self-promotion`), and longer videos (e.g., live streams) or shorter videos (e.g., TikTok videos [164]) may have introduced additional types of information. Further research should explore a wider range of how-to videos, which could build upon our taxonomy.

4.9 Conclusion

We present a taxonomy of information types in how-to videos. Our taxonomy identifies 21 types of information under 8 categories: *Greeting*, *Overview*, *Method*, *Supplementary*, *Explanation*, *Description*, *Conclusion*, and *Miscellaneous*. We demonstrate the utility of the taxonomy in both analyzing users' navigational behavior and supporting their navigation in how-to videos. We first show how our taxonomy can serve as an analytical framework for understanding existing video navigation systems. Then, we further investigate how the information type can assist people watching how-to videos. An explorative user study with nine participants showed that type-based navigation enabled participants to find specific information and perform tasks effectively. We further discuss how the taxonomy enables multiple applications in video authoring, viewing, and analysis. Finally, we release a dataset, HTM-Type, which contains 120 videos containing 9.9k sentences with each sentence labeled according to the taxonomy. We hope that our work builds a foundation for understanding how-to videos in a more systematic way.

Chapter 5. SoftVideo: Improving the Learning Experience of Software Tutorial Videos with Collective Interaction Data

This chapter focuses on the third phase, the Following phase, where learners attempt to follow instructions in the video step by step. In this stage, contextual units such as step difficulty and relevancy help guide the following process. This chapter has adapted and revised content from a paper at IUI 2022 [191]. All uses of “we”, “our”, and “us” in this chapter refer to the coauthors of the aforementioned paper.

5.1 Motivation and Contributions

Tutorial videos provide step-by-step instructions of complex tasks for feature-rich software such as Photoshop [147] and AutoCAD [9]. People watch a tutorial video and try to apply the techniques from the video to their software when learning new techniques [?]. For example, they search for a video about ”removing background from an image” and learn the skill by applying it to their own image.

When following a tutorial video, people often watch instructions and apply them to their own work (e.g., image editing, document editing, video authoring, programming, etc.) by alternating between the video and the software. Commonly, they first watch a step in the video and apply it to their application. If the application results an error or an unintended outcome, users often adjust the pace of the video and rewatch the step, trying to find what they did differently. Most people go through multiple trial-and-error cycles, which could be cumbersome.

Also, when applying instructions from a tutorial video to their software, users need to constantly compare the two to see if they are following correctly. Users can easily miss important details when a demonstration in a video moves too quickly [80], or subtle visual changes are presented in the video [186]. This process is cognitively demanding with constant context switching and is prone to mistakes.

In this research, we propose SoftVideo, a prototype system that helps users plan ahead before watching each step in tutorial videos, gives feedback to users on their progress, and provides help to overcome confusing moments. Users can see step information such as the name of an action or the duration and difficulty of each step to anticipate what is upcoming and prepare, which reduces context-switching overhead. Users also get informed about whether they completed a step or not so that they can be aware of any missed steps. Lastly, users struggling at a particular step can get help suggestions such as slowing down the pace, replaying the step, or seeing relevant steps. SoftVideo detects users’ confusing moments automatically and presents help suggestions at appropriate moments.

To build SoftVideo, we leverage previous learners who had watched the same tutorial and worked toward the same end goal. Collective interaction logs of the video and the software from previous learners can reveal patterns of how people learn from the tutorial. For example, analyzing the logs can detect the steps people frequently struggle in or miss. It can also identify when the user is facing difficulties by comparing their progress to previous learners. Furthermore, it can reveal how people overcome confusing moments, such as by looking at which steps they referred to when completing a step.

We chose Adobe Photoshop as an instance of the software. We collected interaction logs composed of video interactions (i.e., pause, play, jump) in synchronization with Photoshop usages (i.e., actions performed in the software). Collecting interaction data of both sources in a synchronized manner is essential as it captures the actual interaction between the two sources. This allows for more accurate estimations of the user's current task state, enabling SoftVideo to provide appropriate help to people facing the back-and-forth challenges.

We collected 120 complete interaction logs with two tutorial videos (60 logs for each) with 74 participants of varying levels of expertise in Photoshop. Our data analysis pipeline then analyzed the collected data to 1) estimate the difficulty of each step by analyzing how users behaved on each step and 2) identify the relevancy of each step. For 1), we define six measures that portray the difficulty of each step: Execution Time Index, Repetition Time Index, Backjump Frequency, Pause Frequency, Miss Rate, and Re-follow Rate. For 2), we identify the ”Relevant steps” of each step, which are the steps that are performed again in order to complete a particular step.

We evaluated our tool with the two Photoshop tutorial videos with which we collected interaction data. We recruited 30 participants (23 novices, 7 experienced) and asked them to follow a tutorial video with SoftVideo. Results show that participants were able to proactively and effectively plan their pauses and playback speed, and vary their concentration level before watching a step by looking at the presented step information. The difficulty visualization also made them feel relieved when they encountered confusing moments. They were also able to identify and recover from errors with the help SoftVideo provided. Relevant step information helped them overcome confusing moments and acquire contextual Photoshop knowledge.

The primary contributions of this paper are as follows:

- A publicly available dataset of 120 interaction logs across the tutorial videos and Photoshop in use [1].
- SoftVideo, an interface powered by previous users' interaction data that provides step information and real-time feedback to users.
- Results from a study showing that participants used the system to efficiently plan their action and recover from errors in Photoshop tasks.

5.2 Data collection study

	1) Logo	2) Geometry
Outcome		
Effect	Galaxy-style logo design	Geometric Shape Effect
Length	9m 35s	7m 34s
Number of Actions	27	45
URL	youtu.be/ifG1SDxqpAQ	youtu.be/vcLjyGbF40Y

Table 5.1: Tutorial videos used in the data collection study.

In our approach, we leverage interaction logs from previous learners who had watched the same tutorial and worked toward the same end goal. Collective interaction logs of video and the software can

[1] softvideo.kixlab.org

provide useful insights into patterns of how people learn from the tutorial. It can reveal meaningful information of videos, such as where users struggle a lot and thus need to pay attention to. We recruited participants to collect interaction data of both the tutorial video and the software in synchronization. We used Adobe Photoshop as the target software, due to its high availability and popularity. Participants were asked to follow Photoshop tutorial videos and complete image editing tasks.

5.2.1 System for Data Collection

We built a system to collect the interaction data from both the tutorial video and Photoshop synchronously. The system collects video interaction logs (i.e., play, pause, and jump actions with the corresponding video timestamp and user timestamp) in synchronization with software interaction logs (i.e., actions done in Photoshop). In the system, we embedded a Youtube video player for a Photoshop tutorial video. We logged video interaction data using the YouTube player API [7]. To log software interaction logs, we used the History Log feature available in Photoshop. Once users enable the History Log feature in Photoshop, a text file that logs the action history is saved in their local computer. A new line is appended to the file for every action performed in Photoshop. Once a user uploads the path of the text file to our system in the beginning, the system reads the changes in the file periodically and logs the actions in Photoshop, together with the corresponding video timestamp and user timestamp. We stored the logs in Firebase Realtime Database [41].

5.2.2 Participants

We recruited 75 participants from an academic institution through online recruitment postings (48 male, 27 female, mean age 23). We collected their frequency of Photoshop usage on a 5-point scale (1: None, 2: Yearly, 3: Monthly–Yearly, 4: Monthly, 5: Weekly). Based on their responses, we grouped participants who have not used Photoshop or use it 1-2 times a year as novice, and experienced otherwise. We used the frequency of use for grouping expertise because new features are added to the software several times a year [67] and to avoid subjective measures (e.g., self-reported expertise). Each participant completed either one or two tutorials depending on their availability during the given time. The number of collected logs for each tutorial and participants' expertise level is shown in Table 5.2. Participants were compensated with 20,000 KRW (approximately 17 USD) for a 90-minute-long study.

5.2.3 Task

The task was to follow a Photoshop tutorial video about making 1) a galaxy-style logo design ('Logo') or 2) a geometric shape effect ('Geometry') (Table 5.1). We chose the videos from YouTube because they were less than 10 minutes to ensure a feasible study duration, and the tasks were not too trivial (e.g., image cropping) nor too advanced (e.g., poster design).

5.2.4 Procedure

Participants were first assigned to one of the two tutorials. After we introduced the effect and the final outcome of the tutorial, they were asked to prepare images they wanted to use. Participants could optionally choose one of the images we provided. They were then instructed to open Photoshop and our system, and follow the tutorial video. If time allowed after completing one, they followed another tutorial.

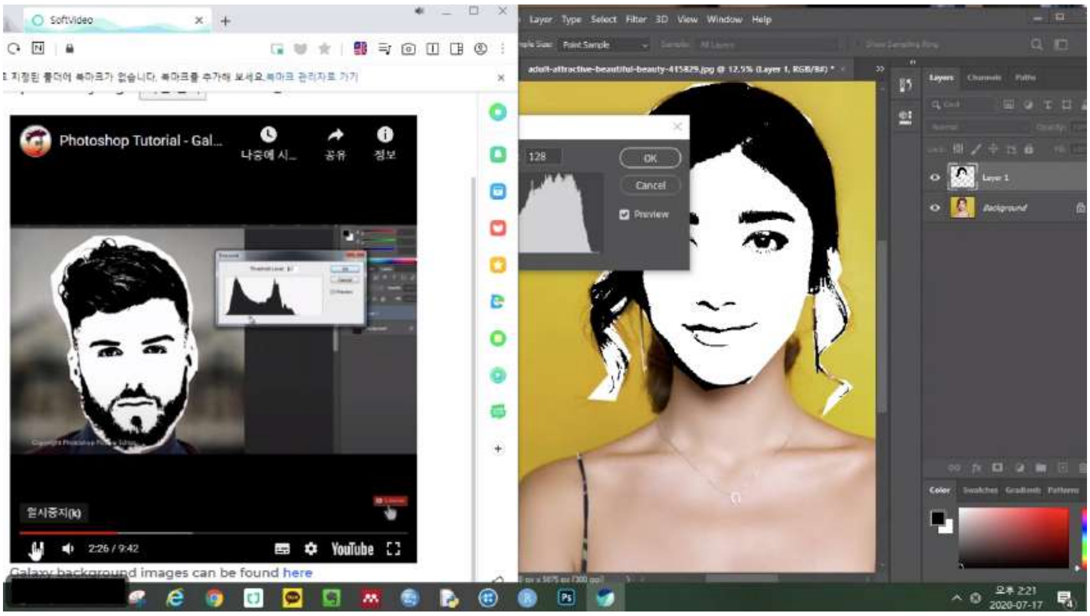

SoftVideo
Photoshop Tutorial - Gal...
나중에 시...
공유
정보
일시중지(k)
2:26 / 9:42
YouTube
Galaxy background images can be found here
Layer Type Select Filter 3D View Window Help
128
OK
Cancel
Preview
Layers Channels Paths
Layer 1
Background
Color Swatches Gradients Patterns
2020-07-17

Figure 5.1: An example session of the data collection study. A participant is following the tutorial video (on the left) on their software (on the right).

The study was conducted in either an offline or online setting. The same system was used in both settings.

- Offline setting: We set up computers with Photoshop installed. We enabled the Photoshop History Log feature and uploaded the path of the log file to our system. A total of 24 participants joined offline.
- Online setting: Participants were asked to install Photoshop and either Whale [174] or Min web browsers [173] before the study to enable real-time tracking of Photoshop usage logs, as other browsers did not support it due to their security policies. They were asked to enable screen sharing during the study. We guided them to enable the Photoshop History Log feature and upload the path of the log file to our system. A total of 51 participants joined online.

5.2.5 Results

With 75 participants, we collected a total of 120 interaction data, 60 for each of the tutorials (Table 5.2). The interaction data is composed of *video interaction logs* and *software usage logs*. Below we specify the scope of the video interaction logs and the software usage logs we collected.

- Video interaction logs: Play, Pause (duration) and Jump (from, to) on the video and the corresponding user timestamps and video timestamps.
- Software usage logs: Actions done on the software (e.g., `Crop`, `Resize`) and the corresponding user timestamps and video timestamps.

The average time taken to complete the tutorial was 32m 54s and 29m 35s for the Logo and Geometry tutorials, respectively (Table 5.3).

	Novice (N=59)	Exp. (N=16)
1) Logo	49	11
2) Geometry	48	12

Table 5.2: The number of collected logs for each tutorial.

	Novice	Exp.	Avg.
1) Logo	35m 24s	21m 46s	32m 54s
2) Geometry	30m 51s	24m 33s	29m 35s

Table 5.3: Average time taken to complete each tutorial.

Measure	**Definition** (*video time*: a duration of a step in video)
Execution Time Index	Time taken to follow a step / *video time*
Repetition Time Index	Total time of a step being watched / *video time*
Backjump Frequency	Number of backward jumps
Pause Frequency	Number of pauses
Miss Rate	The proportion of users who missed a step at first but followed it later
Re-follow Rate	The proportion of users who re-followed a step after proceeded to the next steps

Table 5.4: Definition of six measures that portray the difficulty of each step.

5.3 Data Analysis Pipeline

Our data analysis pipeline analyzes the collected interaction data to identify meaningful information from the tutorial video. Specifically, we aim to 1) estimate the difficulty of each step so that users can plan their action before watching each step, and 2) identify the relatedness of steps so that users can refer to when having difficulties in a particular step. We first describe measures that are used for each of the two purposes.

5.3.1 Measures

Difficulty of steps

We defined six measures that portray the difficulty of each step: Execution Time, Repetition Index, Backjump Frequency, Pause Frequency, Miss Rate, and Re-follow Rate. Table 5.4 shows the definitions of six measures. Below we describe each measure in detail.

- **Execution Time Index**: *(Time taken to follow a step)/(video time).* If a user spends much longer time in a certain step than its length in the video, there is a high chance that the user has difficulties completing the step. For a fair comparison between the steps, we take relative execution time, defined as the time taken to follow a step divided by the video length of the corresponding step. Note that there was no fast-winded or cut parts in the videos we used.

- **Repetition Time Index**: *(Total time of a step being watched)/(video time).* Users repeatedly watch a step if something is unclear from the video or does not work in their context. Similar to Execution Time Index, we take relative repetition time, defined as the total time of a step being watched divided by the video length of the corresponding step. If the Repetition Index is 1.5, the user watched the whole step once, and half of it once more.

- **Backjump Frequency**: (*Number of backward jumps*). Users jump backward on the video to watch the part that is demonstrated quickly or unclearly. We count the number of backward jumps

Measure	Definition
Relevant Steps	Previous steps that users followed after watching the current step to complete the step
Referring Rate	The proportion of users who followed previous steps again to proceed with the current step
Continued Rate	The proportion of users who only watched the current step to proceed with the step (i.e., 1 - Referred Rate)

Table 5.5: Definition of three measures related to relevancy of each step.

that occurred while watching a step.

- **Pause Frequency**: (*Number of pauses*). Users pause the video to transfer the content in the tutorial to their application if it needs much attention. If there are frequent pauses, it may indicate that the step is hard to digest and to be transferred to their context at once. We count the number of pauses that occurred in a step. We do not consider the duration of pauses as it highly overlaps with the Execution Time Index.

- **Miss Rate**: *(Proportion of users who missed a step at first but followed it later).* If a step is not clearly shown in the video, sometimes users skip the step at first. We define the Miss Rate as the proportion of users who missed a step at first but followed it later. A high Miss Rate indicates that users can easily miss the step.

- **Re-follow Rate**: (*Proportion of users who re-followed a step after proceeded to the next steps*). If a step was not completed in the users' context, they might revisit and perform the action again even after they moved on to the later steps. We define the Re-follow Rate as the proportion of users who revisited the step and performed it again. A high Re-follow Rate means many users go back to the step and follow it again, indicating a high chance where the step could not be properly done.

Step relevancy

We defined three measures about relevancy of each step: Relevant Steps, Referring Rate, and Continued Rate. Relevant step information can help learners who get stuck in a certain step, by suggesting they check other related steps again. To help learners decide whether they should check the relevant steps, we also define Referring Rate and Continued Rate. Below we describe each measure in detail (Table 5.5).

- **Relevant Steps**: When users get stuck in a certain step, they sometimes try previous steps again to help them complete the step. We define the previous steps that are followed after watching the current step to complete the current step as Relevant Steps. Figure 5.2 shows an example scenario describing Relevant Steps.

- **Referring Rate**: Referring Rate means the proportion of users who followed the previous steps after watching the current step, to complete the current step. In other words, it is the proportion of users who produced the Relevant Steps. It indicates how relevant the Relevant Steps are.

- **Continued Rate**: In contrast to the Referring Rate, the Continued Rate means the proportion of users who only watched the current step to proceed with the step. In other words, it is (1 - Referring Rate).

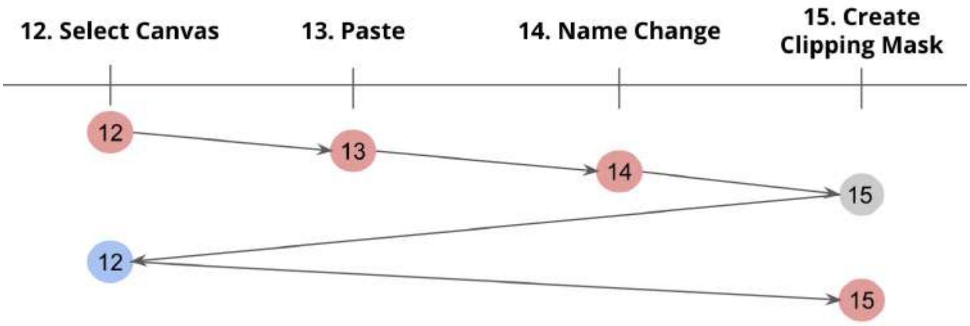


Figure 5.2: An example scenario where the relevant step of step 15 is step 12. After a user followed the step 12, 13, and 14, he is now on step 15. However, the user was not able to complete it. The user jumped back to step 12 and then followed it again. Then, he came back to step 15 and followed the step. (red: followed, gray: watched but not followed, blue: followed again).

5.3.2 Methodology

We describe the methodology we used to compute the above measures for each step from the collected interaction data.

Removing actions that are unrelated to the task

After collecting interaction data—video interaction logs (play, pause, jump backward/forward) in synchronization with the software usage logs—we first processed the software usage logs to remove the actions that are unrelated to tasks. The History Log feature in Photoshop extracts actions done on Photoshop including actions that are not directly related to the main tasks, such as auto-saving files or quitting the application. Thus, we removed log entries that are not related to the tasks.

Identifying the followed and skipped steps

To compute the Execution Time Index, Miss Rate, Re-follow Rate, and Relevant Steps, we need to identify when and which steps were followed or skipped. For example, we need to know when a user successfully followed a step to compute the Execution Time Index.

To identify if a user followed or skipped a step, we first define *baseline actions* as actions done in tutorial videos and *baseline timestamps* as the starting timestamps in the tutorial video of the corresponding baseline action (Figure **??**). To get the baseline actions, we followed the tutorials exactly the same on our Photoshop, checking which action is being logged in the History Log feature. For the baseline timestamp, we manually recorded the timestamp where each action began to be described in the video by watching the tutorial videos.

After setting up the baseline actions and baseline timestamps, we developed an algorithm that detects whether a user followed or skipped a step from the interaction logs (Algorithm 2). The algorithm detects that a user **followed** a step 1) if they performed a baseline action after passing, 2) but still nearby the corresponding baseline timestamp; threshold values in Algorithm 1 determine the range of ”nearby”. The algorithm detects a user **skipped** a step if they did not follow the step but followed the next step. The algorithm detects a user **added** an action if the action does not exist in the video or it exists but is not considered as *followed*.

Algorithm 1: IsFollowed

```
Input:  A list of baseline timestamps, $T = t_0, ..., t_n$
  A list of baseline actions, $A = a_0, ..., a_n$
  A current video timestamp, $t$
  An action performed by a user, $a$
  An index of the expecting action that needs to be done, $i$
  An index of the most recent action that a user has watched, $w$
  Output:  True if the action is a followed action, False otherwise

$thresholdPrevious, thresholdAfter \leftarrow$ Thresholds of video timestamp offsets
  if $i \leq w$ then
    $thresholdPrevious \leftarrow 20$ ;
    $thresholdAfter \leftarrow 20$
else
    $thresholdPrevious \leftarrow 5$;
    $thresholdAfter \leftarrow 15$;
if $a = a_i$ then
    if ( ($i \leq w$ and $a_i$ is unique in $A$) or ($t \geq t_i - thresholdPrevious$ and
     ($t \leq t_i + thresholdAfter$ or $i = n$)) then
        return True
return False
```

Computing the measures for each step

Among the six measures regarding the difficulty of steps, we computed the Execution Time Index, Repetition Time Index, Backjump Frequency, and Pause Frequency for each user per step. Then, we averaged the values among users per step and regarded the averaged value as a representative value of each step. We computed Miss Rate, Re-follow Rate, and the three measures of step relevancy (i.e., Relevant Steps, Referring Rate, and Continued Rate) per step.

To estimate the difficulty of each step, for each of the six measures, we identified the steps with a value higher than the third quartile (i.e., 75%) of all steps. For example, we identified a step with high Execution Time Index by comparing its value to the third quartile of the Execution Time Index values of all steps. We apply the quartile method since it is widely used to classify data into subgroups considering the distribution [18].

Additionally, for the measures that could be computed per user (i.e., Execution Time Index, Repetition Time Index, Backjump Frequency, and Pause Frequency), we computed the third quartile of each measure within a step among users in the same group (i.e., Novice or Experienced). This is to set multiple thresholds to identify if a user is having difficulty. For example, if a novice user’s Execution Time Index

Algorithm 2: Action State Detection

```
Input: A list of baseline timestamps, $T = t_0, ..., t_n$
  A list of baseline actions, $A = a_0, ..., a_n$
  A current video timestamp, $t$
  An action performed by a user, $a$
  An index of the expecting action that needs to be done, $i$
  An index of the most recent action that a user has watched, $w$
  An index of the previous followed action, $p$
  A list of user logs, $L = [(state_0, action_0), ..., (state_m, action_m)]$ ; /* state is either
  'followed', 'added', or 'skipped'                                                  */
Output: $L = [(state_0, action_0), ..., (state_{m+1}, action_{m+1})]$

if $a$ is not in $A$ then
    $L \leftarrow$ L + ($`added'$, $p$);
    return
; /* Check if a user followed the expecting action or previous actions          */
$j \leftarrow i$
 while $j > 0$ do
    if isFollowed(T, A, t, a, i, w) then
        $L \leftarrow L + (`followed', j)$;
        return
    $j \leftarrow j - 1$
; /* Check if a user skipped an action and followed a further action            */
$j \leftarrow i + 1$
 while $j < len(L)$ do
    if isFollowed(T, A, t, a, i, w) then
        for $k = i$ to $j$ do
            $L \leftarrow L + (`skipped', k)$
        $L \leftarrow L + (`followed', j)$;
        return
    $j \leftarrow j + 1$
$L \leftarrow L + (`added', p)$;
```

of a step is exceeding the third quartile of novice users in the same step, we could assume that the user is undergoing difficulty in the step.

5.3.3 Results

Through the analysis, we computed 1) the six difficulty-related measures for each step and for each user per step, and 2) the three step relevancy-related measures for each step. Table 5.6 shows the average values of the six difficulty-related measures across the step for each tutorial. Except for the Miss Rate, the difference between novice and expertise group was statistically significant (Mann-Whitney Test, $p < 0.01$ or $p < 0.05$), showing the reliability of the measures used (Table 5.6). It indicates that novice users showed more behavior of having difficulties than the experienced users.

We describe several examples of the results below. Table **??** shows example measures of steps that exceed the third quartile of all steps, which might indicate that the step is likely to be more difficult. Table **??** shows the third quartile values of each measure for each step, which serve as threshold values when detecting users' confusing moments. Table **??** shows examples of Relevant Steps, Referring Rate, and Continued Rate. We can see that even though steps `Move` and `Select Canvas` from the Logo tutorial all have at least three Relevant Steps, their significance could be different as the Referring Rates differ substantially (41% vs. 8%).

From the analysis, we could also see that Miss Rate demonstrated steps that have certain properties that make them easy to miss. For example, 39% of participants missed the `Drag Selection` on the Logo tutorial, which was passing fast and not noticeable. Re-follow Rate captured steps that need attention. For example, 60% of users followed `Layer Order` again in the Geometry tutorial. Positioning the layers in the right order was important but many participants did it incorrectly at first.

Measure	**Expertise**	**Logo**	**Geometry**	**Avg.**
Execution Time Index	Novice	5.4*	6.3*	5.9
	Experienced	4.0*	5.5*	4.7
Repetition Time Index	Novice	1.82*	1.8*	1.81
	Experienced	1.43*	1.53*	1.48
Backjump Frequency	Novice	1.79**	1.24**	1.52
	Experienced	0.76**	1.10**	0.93
Pause Frequency	Novice	1.60*	1.07*	1.34
	Experienced	1.27*	0.58*	0.93
Miss Rate (%)	Novice	5.1%	4.5%	4.8%
	Experienced	3.2%	5.7%	4.5%
Re-follow Rate (%)	Novice	16.8%*	15.4%	16.1%
	Experienced	8.3%*	13.0%	10.7%

Table 5.6: Mean values of the six difficulty-related measures among all steps. In general, novice users show more behavior of having difficulties than experienced users. For each measure, the table shows if the difference between the novice and experienced groups was statistically significant (*: p¡.05, **: p¡.01, Mann-Whitney Test) for each measure.

5.4 SoftVideo

We present SoftVideo, a prototype system that provides step information, gives feedback to learners on their progress, and provides help to overcome confusing moments (Figure 5.3). SoftVideo provides

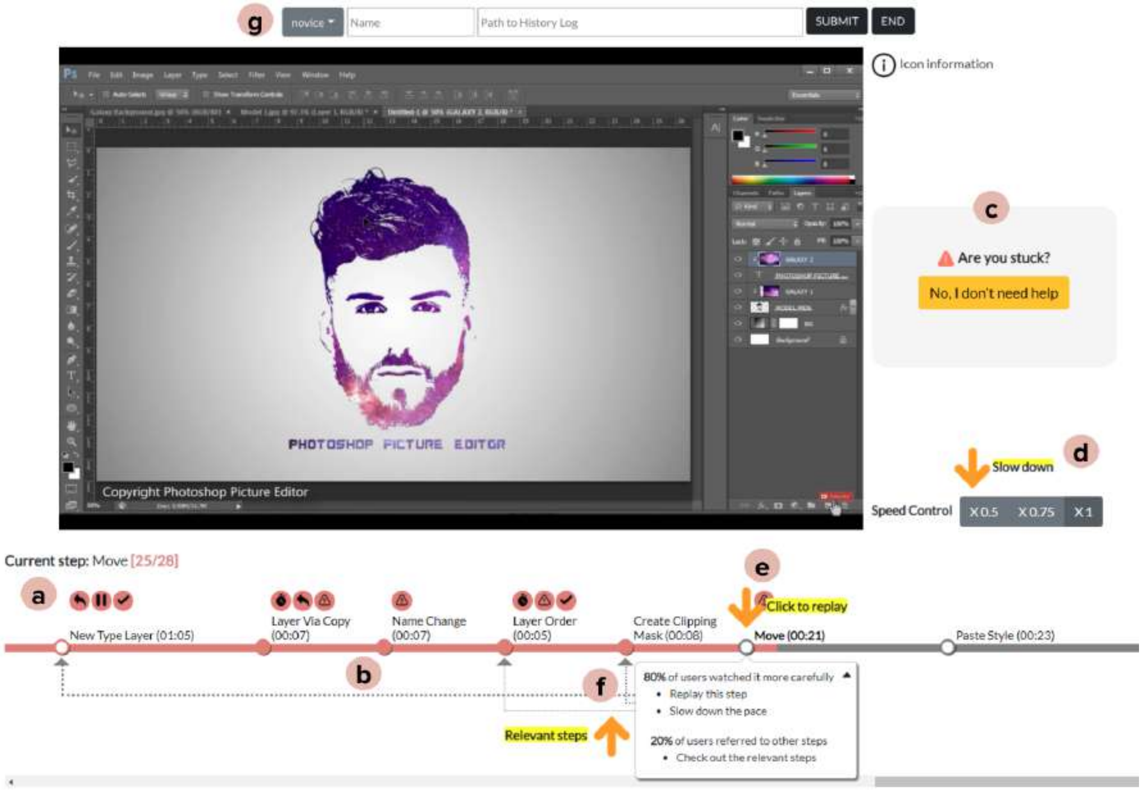


Figure 5.3: Overview of SoftVideo. Along with the software tutorial video, SoftVideo provides (a) a timeline where users can see the action name, its length, and the estimated difficulty. (b) Users can receive real-time feedback on their progress. If a user followed a step, the circle will be filled. (c) SoftVideo detects users' confusing moments. Once detected, it provides users with suggestions such as (d) slowing down the pace, (e) replaying the step, or (f) seeing relevant steps. Users see customized information based on (g) the expertise level they enter.

step information such as the name of an action, and the duration and estimated difficulty of each step in the timeline (Figure 5.3(a)). It gives feedback to users about their progress by letting them know if they completed or missed a step (Figure 5.3(b)) and detecting when they struggle (Figure 5.3(c)). Finally, it presents help suggestions such as to slow down the pace, replay the step, or see relevant steps when they struggle (Figure 5.3(d)-(f)).

There are three components in SoftVideo that are powered by the analyzed data (Section 5.3): Estimated difficulty of each step, criteria for detecting users' confusing moments, and relevant steps that are suggested when they struggle. All the information is determined based on the group the user belongs to (i.e., Novice or Experienced) so that the system provides customized help. User can enter their level of experience before they start watching the video (Figure 5.3(g)).

5.4.1 Step Information

SoftVideo provides a timeline that shows step information in the tutorial video (Figure 1.4). The timeline is segmented into steps and each step is shown with the Photoshop action name and its duration, which is reflected in its length in the timeline. The timeline display of step descriptions has been introduced by other systems (e.g., [84]), but we additionally provide characteristics of each step that represent the

Icons	Meanings	Measures
⏱	Users spent **more time** in this step compared to other steps.	Execution Time Index
🔁	Users watched this step **repeatedly more** than other steps.	Repetition Time Index
↩	Users did **backward jumps frequently** at this step more than other steps.	Backjump Frequency
⏸	Users **paused frequently** at this step more than other steps	Pause Frequency
⚠	There are relatively many users who **missed** the step.	Miss Rate
✓	There are relatively many users who **followed again** the step.	Re-follow Rate

Table 5.7: Icons that depict the difficulty of each step, and their corresponding meanings and measures.

difficulty of a step. With the six difficulty-related measures (Section 5.3.1), SoftVideo presents icons for the measures with values that exceed the third quartile of all steps. Table 5.7 shows the icons and their meanings, and corresponding measures. For example, if a step is shown with the pause icon, it means that users paused frequently at the step more than other steps. Thus, users can estimate the difficulty or complexity of a step by skimming through the icons shown in the timeline. We chose to present such potentially useful indicators rather than a single quantified difficulty level, so that users can have control over how they leverage the given information.

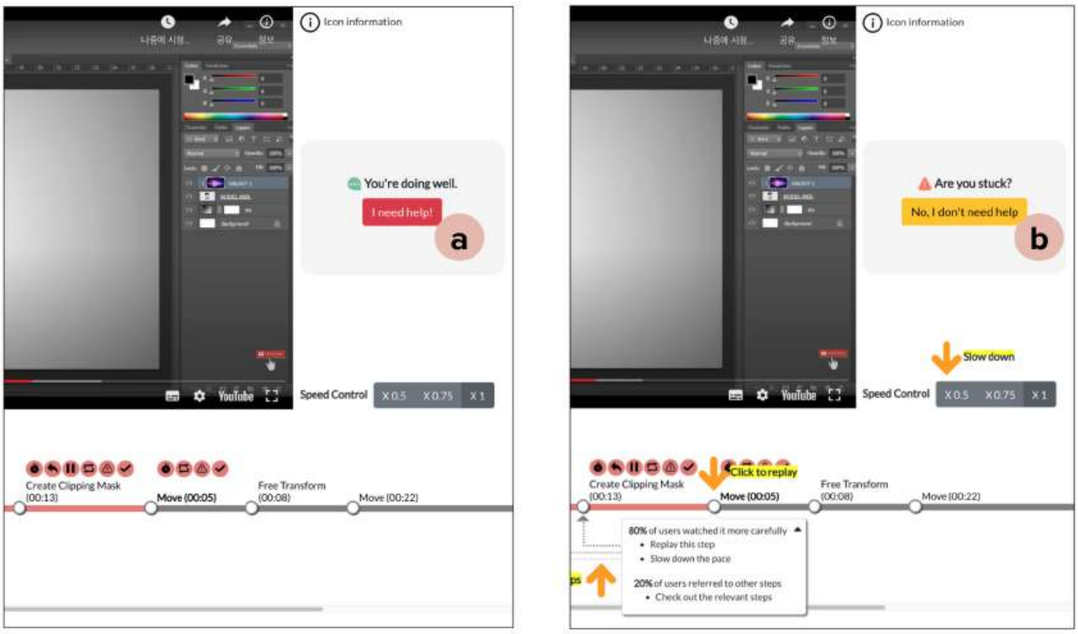
5.4.2 Real-time Feedback

SoftVideo gives real-time feedback to users on their progress by tracking both the video and the application logs. First, it lets users know if they completed a step or not with our action detection algorithm, described in Section 5.3.2. If a user follows a step in their application correctly, then the circle of the step gets filled. If a user misses a step and proceeds to the next step, the circle remains unfilled and the user is warned (Figure 1.4(c)). Second, it detects when a user is facing difficulties. If any of the six measures exceeds its threshold value (Section 29), the system alerts users by asking "*Are you stuck?*" and presents appropriate help suggestions, which are described in the next section (Figure 5.4-right).

5.4.3 Help Suggestions

When SoftVideo detects users undergoing confusing moments, it suggests users to 1) slow down the pace, 2) replay the step, or 3) go back to relevant steps. Users can slow down the video pace to x0.5 or x0.75 by clicking the button (Figure 5.3(d)), or replay the step by clicking the circle on the timeline (Figure 5.3(e)). SoftVideo also suggests users to check relevant steps (Figure 5.3(f)). The arrow to a relevant step is thicker if more users followed the step after watching the current step. To help users better decide if they should check the relevant steps or not, SoftVideo presents the ratio of users who only watched the current step to complete it and users who watched and followed previous steps to complete it (Section 5.3.1). This is to help users with decision making rather than giving pressure to check relevant steps. If a user moves to other steps, the suggested help gets closed.

Users can also request to see help by clicking the "*I need help!*" button (Figure 5.4(a)) or close the help suggestions by clicking the "*No, I don't need help*" button (Figure 5.4(b)). This is to make sure users access necessary help suggestions on demand (or dismiss unnecessary information) in case the algorithm failed to detect their confusing moments.



80% of users watched it more carefully
- Replay this step
- Slow down the pace

20% of users referred to other steps
- Check out the relevant steps

Figure 5.4: (Left) A user is following the tutorial video. Once the system detects that the user may be confused or struggling, (Right) SoftVideo presents action suggestions as help. Users can also proactively (a) request to see the help (b) or close the help.

5.4.4 Implementation

We implemented SoftVideo using React.js, HTML, and CSS for the front-end web interface, and Node.js and Firebase for the backend server. The implementation mostly follows the system used in the data collection study (Section 5.2.1). It additionally runs the action detection algorithm (Algorithm 2) in real-time to track users' progress and runs the data analysis pipeline (Section 5.3) in real-time for computing the Execution Time Index, Repetition Time Index, Backjump Frequency, and Pause Frequency measures to detect users' confusing moments.

5.5 User Evaluation

We evaluated the feasibility of using data-driven information and the effectiveness of SoftVideo through a study. Specifically, the goals of our evaluation were (1) to see how participants think about and use the step information when performing tasks, and (2) to assess the effect of real-time feedback and help suggestions on improving the user experience of software tutorial videos.

5.5.1 Participants

We recruited 30 (22 male, 8 female, mean age 23.8) participants from an academic institution through an online community posting, including 23 novice and 7 experienced users for Photoshop. The level of Photoshop expertise were determined in the same manner as in Section 5.2.2. People who participated in the data collection study were excluded from this recruitment. Each participant was assigned to one of the two tutorial videos used in the data collection study. We assigned the participants equally for each tutorial; 15 (11 novice, 4 experienced) were assigned to the Logo tutorial while the remaining 15 (12 novice, 3 experienced) were assigned to the Geometry tutorial. Participants were compensated 20,000 KRW (approximately USD 17) for their participation in a 80-minute-long study.

5.5.2 Study Procedure

The study took place face-to-face, following the COVID-19 guidelines: participants had to wear masks and sanitize their hands before using computers. Windows and doors were open and an air conditioner was turned on to keep the room ventilated. We sanitized the utilities after each session.

Participants were first asked to complete a pre-task survey about their experiences in using Photoshop and how they interpret each of the six message types (Table 5.7) to make sure they become familiar with the messages. We then introduced a Photoshop tutorial video to participants and asked them to choose images to be used based on their preference. After explaining how to use SoftVideo, one researcher set up their expertise level (novice or experienced) in SoftVideo based on the pre-task survey result and entered the path to the Photoshop History Log file for real-time tracking. Participants were then asked to follow the given tutorial video using SoftVideo. Once participants completed the main task, we conducted a survey about their experience and a semi-structured interview to get more detailed feedback. Each participant was provided with two monitors; one for the tutorial video (SoftVideo) and the other for Photoshop.

We chose not to do a comparative study as SoftVideo is a complex system with multiple novel features: a comparative study cannot clearly uncover the source of differences observed, and it is unclear what a convincing baseline might be. Rather, we focus on observing and analyzing how participants use SoftVideo in a realistic task. We logged the number and the timestamp of detected confusing moments, help requests and help dismissals made by participants, and their usage of help suggestions.

5.6 Results

Below we summarize the main findings and usefulness of SoftVideo with respect to each feature.

5.6.1 Step Information

Participants were able to estimate the difficulty of steps with the number of icons shown in the timeline. In general, they felt that the number of icons implied the difficulty of a step (perceived accuracy = 3.73/5, std=0.98). Being able to know about the difficulty of steps affected them in a few different ways, which we report below.

Participants planned their behavior and level of concentration according to the difficulty of steps.

Participants were able to plan their action and level of concentration by looking at the difficulty of upcoming steps. They planned their pauses on the video depending on the difficulty (P3, P4, P22, P26). P22 said, *"I put my fingers on the space bar in advance when facing difficult steps so that I can be ready to pause."* Similarly, P3 said, *"when there were no icons, I tried to watch the step at once until the end without pauses."* Participants not only planned their pauses but also controlled the speed of the video playback (P1, P10, P14, P23). P1 said, *"I was able to prepare myself for upcoming steps by slowing down the pace whenever I saw many icons."* Even if they did not perform an explicit action to be prepared, they adjusted their level of concentration based on the difficulty (P11, P13, P17, P19, P23, P24, P27, P29).

P13 said, *"When there were no icons, I was relaxed and watched the step in a relaxing way. However, when there were many icons, I focused more."*

Participants' experiences in early steps affected their planning strategy. P14 said, *"I found myself being able to watch and follow at the same time when there were two or fewer icons. After experiencing that I pause a lot during steps with three or more icons, I started to slow down the pace of the video right before such steps came up."* P6 built their own understanding of the icons through the earlier steps which made them perform certain actions prior to watching steps with particular icons. P6 said, *"I learned that there was a pause icon whenever the step required me to enter in some parameters like width and height. After experiencing it, I was able to know when similar actions (i.e., setting values) are coming (when I saw the pause icon) and so I was able to perform them in advance."*

Step-wise difficulty information increased the level of safety and gave hints when they struggle.

When participants faced confusing moments, they checked to see icons and felt relieved to see many icons on the step (P4, P5, P8, P12, P16, P18, P19, P27). P27 said, *"I felt relieved to see many icons when I was struggling because I knew it was not only me and the problem is the step itself."* It also happened when participants came back to a certain step after having done it differently or missed it. P18 said, *"I didn't notice the icons at first, but when I revisited a step to do it again, I could see many icons and was able to know that there were many similar users like me."*

The difficulty level also gave hints on how to overcome confusing moments—whether they should look into the step in more detail or watch other steps. P7 said, *"When I struggled, I watched the step more carefully if there were many icons. In contrast, if there were few icons, I realized something went wrong in previous steps, not the current step, so I watched previous steps."*

Differences in the perceived usefulness between the messages

Although most participants perceived the icon count as an indicator of step difficulty, there were differences in perceived usefulness between the messages. Participants rated the usefulness of messages as follows (ordered by score): Pause Frequency (4.03/5), Repeat Index (3.73/5), Revisited Rate (3.73/5), Execution Time (3.63/5), Backjump Frequency (3.6/5), and Missed Rate (2.7/5). Pause Frequency might have been the most useful because knowing how to split a step is important in following tutorial videos. P27 said, *"I tended to pause if there was the pause icon when I wasn't sure about when to pause."* On the other hand, Missed Rate might have been the least useful because participants might have felt that there are small chances of missing a step, partially due to SoftVideo's feature of letting users know if they have missed a step. In general, participants said it was helpful to see step information (3.7/5, std=1.3).

5.6.2 Real-time Feedback

We report how participants felt about real-time feedback on their progress and automatic detection of confusing moments.

Letting users know about their progress

With the feedback SoftVideo provides, participants were able to identify missed steps (P4, P6, P9, P18) as well as steps that they performed differently from the tutorial video (P7, P12, P14). P9 said, *"I noticed a difference between the image on the tutorial and the image on my application. Then I noticed* *that there was a step that I missed due to an alert SoftVideo gave. I was able to go back to the missed step and follow it."* SoftVideo also let users know about steps that they thought they followed but not actually because they behaved differently. P7 said, *"I thought I followed the step `Move` but it didn't appear to be so, so I checked it again. I realized that I didn't press 'Ctrl' while doing the action."* Participants mentioned that the real-time feedback on the progress encouraged them to follow the tutorial more meticulously (P11) and it made following along more enjoyable as it felt like solving a series of quests (P25). Participants said the feature was helpful in general (3.67/5, std=1.3).

Detecting confusion moments

Overall, participants felt that SoftVideo detected their confusing moments accurately. On a scale of 1 to 5, with 1 being early and 5 being late about the timing of SoftVideo's confusion detection, participants rated 2.9 (better if closer to 3, std=0.92). P6 said, *"I thought it detected quite well. I was struggling at a step of doing 'Ctrl+T' and the system detected it right away."* On average, SoftVideo detected 17.83 confusing moments per user (min: 1, max: 29). Participants closed 2.76% of the suggested help and requested to see help 0.77 times additionally on average. For about 32% out of 543 detection and requested cases, participants utilized at least one of the suggested help, which we discuss next.

5.6.3 Help Suggestions

We report the usage of help suggestions by SoftVideo and how participants found information of relevant steps helpful.

How participants used suggested help

Among the three help suggestions SoftVideo provides (i.e., speed control, repeating a step, and relevant steps), participants repeated a step most frequently (114), followed by checking the suggested relevant steps (46) and slowing down the pace (13). Participants might have repeated a step a lot because it is what most users are familiar with, checking if they have missed anything and figuring out why it does not work on their application by watching over and over. On the other hand, they rarely slowed down the pace when faced with difficulties. P25 said, *"I didn't use the speed control because the part that needs attention only lasted a few seconds. I didn't want it to be slower for the entire step."*

How seeing relevant steps was helpful

Participants reported that seeing the suggested relevant steps was helpful in overcoming confusing moments (P2, P5, P7, P9, P11, P14, P16, P19, P23). It helped them by suggesting steps that they should watch again. P2 said, *"When I knew I made a small mistake, I jumped back to 5 seconds before by using the left arrow key on the keyboard. However, when I wasn't sure what caused a problem, seeing the relevant steps was helpful."* In particular, if one of the relevant steps was pointing to a step that they have missed, they perceived it as an important step and went back to the step to follow it (P5, P7, P14, P19). It not only helped participants follow the step they have missed, but also to re-follow the step that they have followed before. P11 said, *"I was able to catch up right away after watching a relevant step. Even though I followed the step, there was something I pressed in a wrong way."*

Some participants perceived the relevant steps as "similar steps", and transferred the knowledge of the step to the current step. P8 mentioned *"I was able to relate the information from a relevant step. I remembered how I completed the step, so I thought I could do this step in a similar way."* Another interesting usage was that it helped participants acquire the knowledge of the software, by looking at which steps are frequently related. P25 said, *”It helped me a lot in understanding how to use Photoshop in general. I was able to know which actions are related and which should be done for other actions to be done.”* Also, with relevant steps participants reported feeling safe because even if they failed to follow a step, there are alternatives that they could try (P26, P27).

However, unlike our expectations, the Referring Rate and Continued Rate were rarely used. Nearly all participants mentioned that they did not look at the numbers. P7 said, *”I didn't see the numbers at all. If there was at least one relevant step, I checked it out no matter how many referred to it.”* Although the Referring Rate and Continued Rate were not used in deciding whether they should watch relevant steps, some participants used the information to adjust their concentration level on the relevant steps (P2, P24). P24 said, *”If the Referred rate was about 80%, I watched it normally. If it was higher than 85%, I paid more attention. If it was 92% or higher I paid extra attention and watched it carefully.”*

5.6.4 Other Feedback

Participants also appreciated the basic timeline that shows the name and duration of each action. It helped them learn about the sub-goal of each step (P4, P8, P17) and made it easier to navigate the video (P1, P9, P13, P29). Seeing the action name was helpful because participants were able to expect which menu they should click (P29), especially when the same step appears again later (P19). Overall, participants found SoftVideo helpful in following along the tutorial content (4.17/5, std=0.87). Moreover, they preferred using SoftVideo compared to the basic video-only interface (*‘I'd prefer to use this system to the basic video-only interface.’* (5-point Likert scale): 4.13/5, std=0.97).

5.7 Discussion, Limitations, and Future Work

In this paper, we investigated the feasibility of enhancing software tutorial videos with data-driven information. In this section, we discuss considerations, limitations, and possible future work of using collective interaction data.

Utilizing Synchronized Interaction Data of Both Software and Tutorial Video

Synchronized interaction data of how a user uses both the software and the tutorial video possess much more potential than just two single data sources. It allows for more accurate inference of the user's current state and more personalized support. For example, our Execution Time, Missed Rate, and Revisited Rate measures are induced from (and are only made possible by) synchronized data of both the software and the tutorial. Using such metrics extracted from synchronized data, in addition to metrics obtained from video interaction logs (i.e., Repetition Index, Backjump Frequency, and Pause Frequency) which have been shown to be relevant with video difficulty [98], we were able to detect whether the user is experiencing difficulty in following the tutorial. Similarly, previous work also showed that utilizing additional logs such as physiological data collected from smartwatches can significantly improve the video difficulty detection [38]. Likewise, if we only utilized one data source or if the data was not synchronized, the impact of SoftVideo could have been less significant.

Users' Trust and Interpretations on Data-driven Information

SoftVideo's data-driven information shows the collective behavior of a number of users who have worked toward a shared goal. How users perceive the meaning of information might be different from user to user. Participants from our study built up their trust towards the system and came up with their own understanding of how to interpret the provided information as they used the system. P26 said, *"I found out that those steps do not have icons because I could easily follow the video while watching at the same time."* Similarly, P4 said, *"It was cool that I actually paused a lot in steps with many icons."* This shows that their trust towards the system grew as they used the system and their experience aligned with the presented information. After understanding how the presented information matches their context, participants built their own techniques to interpret and follow subsequent steps (e.g., to pause the video at steps with three or more icons). It also shows that giving users control to selectively leverage useful signals rather than presenting a single answer predicted by the system allowed them to build trust and make their own interpretations.

Availability of Interaction Data and Its Privacy Implications

In order to utilize synchronized interaction data of both software and tutorial video, it is essential to first consider how to obtain software interaction data. For example, our work uses Photoshop as an instance of software, which enables tracking software usage logs through its History Log feature. Modern software applications such as AutoCAD [9] or Fusion360 [1] also provide history logs so that users can track their progress and easily revert to a particular action. For software with no history logs or API for them, accessibility APIs [58, 113] or computer vision techniques [148, 13, 112, 105] could be used to reverse-engineer the software interactions. Augmenting open-source software such as GIMP [72] could be another possible solution.

When capturing interaction data, privacy issues should be carefully considered. Unlike videos that are published publicly on online platforms, the software is often where users work privately. Previous work suggests that when users acknowledge that there are enough benefits provided, users' perceived privacy concerns may be alleviated [85, 146], but still sensitive personal information or assets (e.g., file names) can be recorded in the software usage logs. Potential solutions include automatically filtering out such information or giving users control by allowing them to review and filter what gets shared.

Leveraging Richer Interaction Data

In our work, we collected pause, play, and jump as video interaction data and Photoshop action names as software interaction data. Future work could look into leveraging richer interaction data. For example, playback speed change or volume control of videos might capture important or non-important parts of the video. Also, users' Undo and Redo behavior on the software can be used [126, 49], as it may imply important moments of the video such as confusing parts or the parts where people explore. With such data, it may be possible to identify steps that are optional or steps where users can branch out and be more creative about. As such, more extensive interaction data could improve the accuracy in revealing important points in tutorial videos. Moreover, analyzing the interaction data with respect to users' expertise level or quality of outcome can enable tailored support according to expertise level or goals.

More Support for Learners and Authors of Educational Videos

SoftVideo demonstrates how utilizing interaction data can enhance the learning experience of software tutorial videos. Extending this idea, future systems can provide further support to learners. As people use the system, the system can give adaptive information to users. The system can control the amount and the content of the information in a personalized way by identifying what information a user needs. For example, a certain part of the video can be only shown to users who encounter a certain type of difficulties. Also, although we set the third quartile as a universal metric when defining the difficulty of a step or detecting users' confusion, future work can investigate adaptive techniques for identifying the user's state and providing more personalized experiences.

Furthermore, our system could be beneficial for authors of educational videos. For example, an author of an instructional video can identify where users struggle a lot or which steps users miss frequently so that they can improve the video or provide additional explanations. Visual analytics tools of how users learn through instructional videos might give insights into understanding users and improving the content as well.

With our public dataset of synchronized interaction logs of the tutorial videos and the software, we expect that it could facilitate a further understanding of how users learn from software tutorial videos. We expect that it will enable future research in data-driven video-based learning.

5.8 Conclusion

This paper presents SoftVideo, a data-driven interface for improving the learning experience of software tutorial videos. SoftVideo helps users plan ahead before watching a step, gives feedback on their progress, and presents help suggestions when they struggle. We analyzed collective interaction logs of a tutorial video in synchronization with the software to provide the difficulty of each step, detect users' confusing moments, and suggest relevant steps. A user study showed that data-driven information allowed participants to plan their behavior of following the tutorial, feel relieved, and overcome confusing moments. We believe that leveraging richer interaction data could further enrich the learning experience of both instructional videos and complex software.

Chapter 6. GUIDE: A Benchmark for Understanding and Assisting Users in Open-Ended GUI Tasks

This chapter focuses on the final phase, the Autonomous phase, where users work independently in their own environment. In this stage, user behavior states and intent serve as useful contextual units for understanding user demonstration videos and providing appropriate assistance.

6.1 Motivation and Contributions

Graphical User Interface (GUI) agents hold great promise for supporting users in complex workflows, in mobile [108, 75, 199], web [159, 68, 193, 43, 206], and software application tasks [197, 137]. In creative and analytical tools such as Photoshop or PowerPoint, these agents can automate repetitive subtasks or provide guidance to help users achieve their goals more efficiently. Most existing GUI agents, both in academic research [61, 104, 201] and in commercial services like Microsoft Office Copilot [118] or Figma Make [52], focus on full automation: given a goal, they either execute a sequence of clicks and keystrokes to complete the task or directly generate the desired output. While this approach offers convenience, it overlooks how people actually work with software. In real-world open-ended creative or analytical workflows, users often prefer to retain control—to experiment, explore alternatives, or iteratively refine their designs [78]. An agent that takes over the entire interface can undermine the user's agency and may even slow down the progress when users must repeatedly revise prompts or undo automated actions.

Recent work on proactive task assistance takes a more balanced approach [177, 111, 183, 198, 184]. Rather than automate tasks for users, proactive assistants infer a user's context and intent and deliver timely, relevant help. Studies in programming and productivity tools show higher efficiency and satisfaction when a system detects a need and intervenes at the right moment [151, 28, 144, 177]. Yet, the ability to model and track users' evolving behavioral context remains underexplored in current multimodal systems that power GUI agents.

To achieve a truly human-assisting GUI agent, a key ability is to comprehend users' cognitive context and intentions to provide appropriate support [69]. In real-world scenarios, users rarely articulate their goals or needs explicitly, making it natural for systems to rely primarily on visual cues from the screen. These user actions often carry semantic structure, such as hovering, undoing, or repeatedly opening menus, that signal intent. However, interpretation remains challenging: similar actions may stem from entirely different intents. For example, repeated undo actions might indicate confusion or deliberate refinement. As a result, without deeper reasoning, assistance based solely on surface-level actions can lead to shallow or misaligned responses.

To address this challenge, we present **GUIDE** (**GUI U**nderstanding, Intent, and Help **D**ecision **E**valuation), a benchmark designed to evaluate multimodal models (MLLMs) on their ability to understand and assist users in complex software workflows. GUIDE introduces a three-stage evaluation framework: (1) *Understanding* the user's behavioral state to identify their current workflow phase; (2) *Reasoning* about their underlying intention and what they aim to accomplish; and (3) *Assisting* by delivering the appropriate form of help at the right moment.

We collected 67.5 hours of screen recordings from 120 human demonstrations across 10 widely used applications—including Photoshop, Figma, PowerPoint, Premiere Pro, and Excel—covering 40

Dataset	Domain #	Video #	Video Duration	Video Source	Primary Goal	Evaluation Focus		
						Behavior	Intent	Help
PsTuts [95]	1	-	71.4 h	Instructional Videos	Action Understanding			
VideoWebArena [74]	6	74	3.8 h	Human-Recorded Tutorials	Task Automation			
VideoGUI [104]	11	178	7.1 h	Instructional Videos	Task Automation		✓	
UI-Vision [128]	83	450	4.8 h	Experts Performing Tasks	Task Automation			
AssistGUI [61]	9	100	<8.3 h	Instructional Videos	Task Automation		✓	
WorldGUI [201]	10	611	<30.5 h	Instructional Videos	Task Automation		✓	
GUIDE (Ours)	10	120	67.5 h	Novice Demonstrations	Behavior Understanding	✓	✓	✓

Table 6.1: Comparison of GUIDE with existing GUI video understanding datasets. GUIDE differs from existing benchmarks by (*i*) collecting screen recordings from novice users, (*ii*) capturing how they naturally behave in open-ended tasks with a focus on behavior understanding, and (*iii*) evaluating systems based on human user needs rather than task automation.

open-ended tasks designed to elicit natural user behavior. Unlike prior work that primarily targets video understanding from expert-recorded instructional videos on closed-ended tasks [95, 104, 128, 61, 201], our focus is on novice users working on open-ended tasks, with the goal of building collaborative AI systems that assist users during exploration, trial-and-error, and learning. Observing novice workflows allows us to capture authentic moments of confusion, decision-making, and discovery, offering rich opportunities for AI to provide timely, context-aware support. Each session includes both screen recordings and think-aloud narrations that surface the user's underlying intentions and cognitive states.

Building on this dataset, we define three-staged benchmark tasks: First, **(i) Behavior State Detection** evaluates whether a model can identify the user's behavioral state, such as exploration or confusion, based solely on visual cues. To support this, we developed a taxonomy of nine user states reflecting diverse cognitive and behavioral phases in open-ended GUI workflows, grouped into four high-level categories: Planning, Execution, Problem-Solving, and Evaluation (Figure 6.4). This structure aligns with human cognition and interaction theories [17, 131], while introducing finer distinctions tailored to GUI-based task behavior. Next, **(ii) Intent Prediction** targets inference of the user's immediate goal—what they are trying to accomplish in the given moment. The final task, **(iii) Help Prediction**, assesses whether a model can determine 1) whether the user needs assistance or not, and if so, 2) what type of help would be most appropriate, such as explaining a feature, suggesting an alternative, or addressing an error. By leveraging both visual screen recordings and accompanying think-aloud narrations, we automatically generated data for each task, which was subsequently verified through human review for accuracy and consistency.

Evaluation across eight state-of-the-art MLLMs reveals that while current models struggle to interpret user behavior and predict underlying intent and help needed—achieving only 44.6% accuracy on behavior state detection and 55.0% on help prediction, performance improves significantly when structured user context is provided. For example, supplying behavioral state and intent information boosted help prediction accuracy by up to 50.2 percent point for the lowest-performing model.

Our results suggest a promising path forward: providing different layers of human-grounded context, such as behavioral cues and inferred goals, can lead to more accurate assistance decisions. These findings indicate that training models on data reflecting users' behavior, intentions, and help needs may enable agents to reason more deeply and assist more effectively. To facilitate future research on collaborative GUI agents, we will publicly release the dataset.

6.2 GUIDE Benchmark

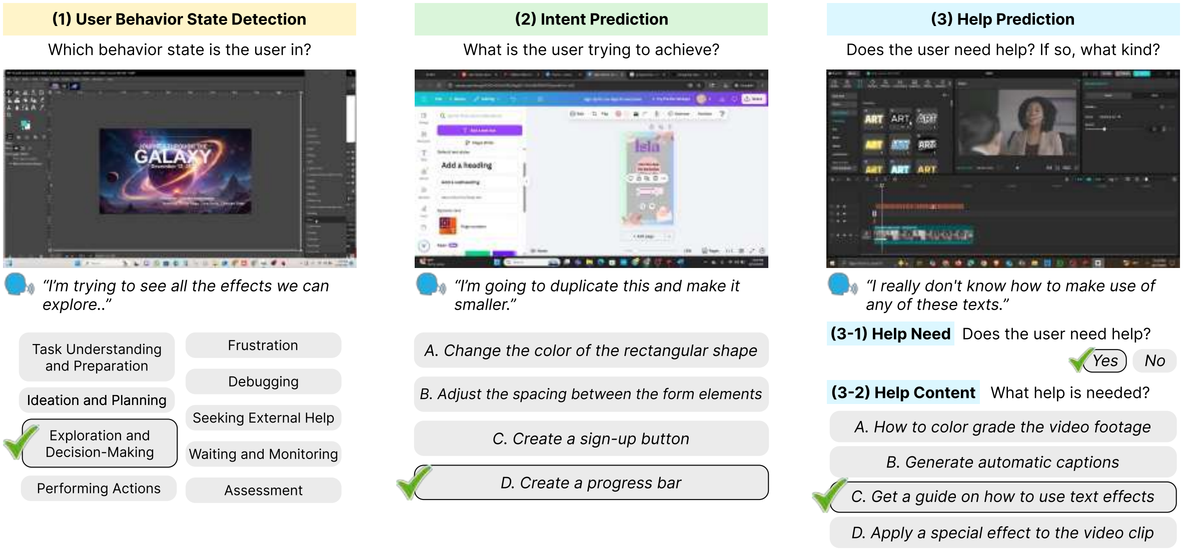

(1) User Behavior State Detection
Which behavior state is the user in?
"I'm trying to see all the effects we can explore.."
Task Understanding and Preparation
Ideation and Planning
✓ Exploration and Decision-Making
Performing Actions
Frustration
Debugging
Seeking External Help
Waiting and Monitoring
Assessment
(2) Intent Prediction
What is the user trying to achieve?
"I'm going to duplicate this and make it smaller."
A. Change the color of the rectangular shape
B. Adjust the spacing between the form elements
C. Create a sign-up button
✓ D. Create a progress bar
(3) Help Prediction
Does the user need help? If so, what kind?
"I really don't know how to make use of any of these texts."
(3-1) Help Need Does the user need help?
✓ Yes
No
(3-2) Help Content What help is needed?
A. How to color grade the video footage
B. Generate automatic captions
✓ C. Get a guide on how to use text effects
D. Apply a special effect to the video clip

Figure 6.1: Overview of the three core tasks in the GUIDE benchmark. (1) **User Behavior State Detection** identifies the user's current behavioral mode (e.g., *Exploration and Decision-Making*). (2) **Intent Prediction** infers what the user is trying to achieve (e.g., *Create a progress bar*). (3) **Help Prediction** determines whether the user needs assistance and, if so, what kind of help is relevant (e.g., *Get a guide on how to use text effects*). Together, these tasks enable a comprehensive understanding of user behavior and assistance needs in software GUI environments. We evaluate MLLMs on their ability to infer these solely from the visual input, without access to the demonstrator's narration — a setting that closely reflects real-world use.

To develop a benchmark that focuses on understanding and assisting users, we collected demonstrations from novice users. Unlike existing datasets that focus primarily on expert demonstrations or polished instructional videos [95, 104, 128, 61, 201], our dataset captures the authentic challenges and exploratory behaviors that novices exhibit during task completion, serving a crucial role in building collaborative agents. Building on these demonstrations, we propose a suite of tasks designed to evaluate models' capabilities to understand users and provide effective assistance.

6.2.1 Video Collection

We collected 120 demonstrations from novice users across 10 software applications spanning five categories: Photo Editing (Photoshop, GIMP), Graphic Design (Figma, Canva), Presentation Design (PowerPoint, Google Slides), Video Editing (Premiere Pro, CapCut), and Data Analysis (Google Sheets, Microsoft Excel). For each application, we designed four open-ended tasks aimed at eliciting natural and diverse user behaviors and approaches (Table **??** in supp.).

We chose creative and analytical tools to surface exploratory workflows and variation in problem-solving strategies. Each task was completed by three different users to capture diverse strategies and

Variable	Value
# Videos	120
Total Duration	67.5 hours
Avg. Duration	33 min 44 sec
Max Duration	1 hour 23 min 50 sec
Min Duration	16 min 42 sec
Think-Aloud Narration Ratio	78%
Task Samples & Granularity	
(1) Behavior State Detection	1.8K
Avg. Segment Length	14.16s
(2) Intent Prediction	1.3K
Avg. Segment Length	25.40s
(3) Help Prediction	1K
Avg. Segment Length	25.56s

Figure 6.2: Statistics of the GUIDE dataset.

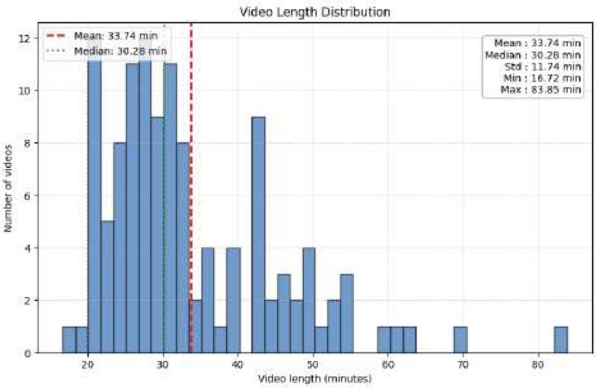


Figure 6.3: Distribution of screen recording video lengths.

behaviors. We ensured that each task was flexible enough, while still incorporating elements of challenge. Participants were asked to spend at least 20 minutes per task and meet a few minimal requirements (e.g., inserting a relevant image) to mark it as complete.

We recruited 54 novice users of software from Prolific and our institution. Participants were screened based on their self-reported expertise and familiarity with the features in each application to ensure they were novice users. During the study, participants worked on the assigned task while recording their screen and keyboard/mouse input events. They were also asked to think aloud and record their voice as they carried out the task, verbalizing what they were doing and their thought process.

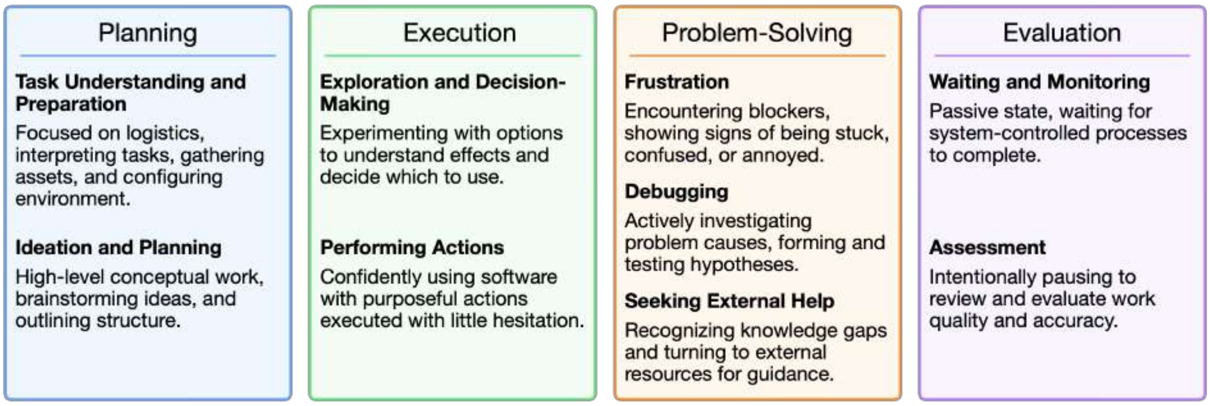
6.2.2 Benchmark Tasks

To evaluate a model's ability to understand user context and deliver appropriate assistance, we design our benchmark as a unified three-stage framework: *Understanding* → *Reasoning* → *Assisting*. These stages progress from interpreting user behavior to inferring intentions and ultimately providing helpful assistance. Each task corresponds to a distinct level of cognitive inference required for a human-assisting GUI agent to effectively support users in open-ended software workflows.

To construct a dataset for task evaluation, we used the Human-AI collaborative method. We first transcribed the think-aloud narration using WhisperX [11], and used the narration as a main source of extracting initial annotations in addition to the video. We employed *Gemini-2.5-Pro* to first create annotations needed for each task, which were then refined by human annotators. Note that we use narration only as an annotation source to capture users' intentions and mental states. The benchmark evaluates vision-only understanding, testing whether models can infer these states solely from visual cues, as in real-world settings without access to user speech.

User Behavior State Detection

Description. This task evaluates whether a model can interpret the user's behavioral context directly from visual cues. Models are asked to classify a video segment into one of nine behavior states in our taxonomy (Figure 6.4), which spans the full range of cognitive and behavioral processes observed in creative and analytical workflows. Grounded in established theories like Norman's Human Action



Planning

Task Understanding and Preparation
Focused on logistics, interpreting tasks, gathering assets, and configuring environment.

Ideation and Planning
High-level conceptual work, brainstorming ideas, and outlining structure.

Execution

Exploration and Decision-Making
Experimenting with options to understand effects and decide which to use.

Performing Actions
Confidently using software with purposeful actions executed with little hesitation.

Problem-Solving

Frustration
Encountering blockers, showing signs of being stuck, confused, or annoyed.

Debugging
Actively investigating problem causes, forming and testing hypotheses.

Seeking External Help
Recognizing knowledge gaps and turning to external resources for guidance.

Evaluation

Waiting and Monitoring
Passive state, waiting for system-controlled processes to complete.

Assessment
Intentionally pausing to review and evaluate work quality and accuracy.

Figure 6.4: Our proposed taxonomy of user behavior states in GUI-based software tasks, organized into four main phases: **Planning**, **Execution**, **Problem-Solving**, and **Evaluation**. Each phase captures distinct patterns of user cognition and interaction, from initial goal formulation to iterative action, troubleshooting, and reflection.

Cycle [131] and Bloom's Taxonomy [17], our taxonomy provides a structured foundation for understanding user behavior, from early planning to problem-solving and reflection.

We developed the taxonomy through a multi-stage, human–AI collaborative process [93]. First, three authors iteratively created and consolidated an initial taxonomy over five sessions based on observations of online software task videos. Separately, we prompted *Gemini-2.5-Pro* to generate a taxonomy from scratch using our collected video dataset, without providing our initial version. We then augmented the human-generated taxonomy by integrating novel categories identified by the LLM. Finally, the combined taxonomy was validated against the entire video dataset to ensure comprehensive coverage and reorganized into the final set of nine distinct states.

Dataset Curation. After constructing the taxonomy, we aligned each video with its corresponding narration segments. For every segment, we annotated the user's behavior state using *Gemini-2.5-Pro* according to the taxonomy, prompting the model to produce both a predicted label and its reasoning. Two human annotators recruited from Prolific then verified and refined these annotations, achieving a 96.1% agreement rate. Finally, we uniformly sampled 200 instances from each of the nine classes, resulting in a balanced dataset of 1.8K annotated segments.

Intent Prediction

Description. This task evaluates whether a model can reason about the user's short-term, immediate goal in context. It focuses on identifying what the user aims to achieve within open-ended workflows.

Dataset Curation. Using the narration-aligned video segments, we prompted *Gemini-2.5-Pro* to infer users' intention in each segment. The think-aloud narrations often revealed users' goals (e.g., *"I'm going to align these objects", "I'll try another color"*). Leveraging this signal, we prompted the model to infer the underlying user intention. After collecting and deduplicating the inferred intents, we further instructed the model to generate three plausible but incorrect alternatives to serve as distractors for the multiple-choice evaluation. The resulting intent annotations and distractors were then validated by the authors, with 88.68% of the data retained, yielding a final set of 1.3K instances.

Help Prediction

Description. The final task evaluates whether a model can progress from understanding and reasoning to deciding how to assist. Help Prediction consists of two subtasks: (1) **Help Need Detection**, a binary classification task that determines whether the user needs help, and (2) **Help Content Prediction**, which identifies the specific type of help needed—such as explaining a feature or suggesting an alternative. Together, these subtasks assess a model's ability to anticipate user needs and recommend appropriate assistance, bridging the gap between perception and actionable support.

Dataset Curation. We identified potential help-seeking moments using two complementary signals. First, *explicit help-seeking* behaviors, such as switching to external resources (e.g., Google, YouTube, ChatGPT) indicated direct attempts to seek guidance. Second, *implicit help-seeking* cues were extracted from user narration, where they expressed uncertainty or confusion (e.g., *"How do I align this?"*, *"I can't find Layer Mask."*). Additionally, we included clear *no-help-needed* moments, where users demonstrated confidence through their narration. Using these signals, *Gemini-2.5-Pro* was prompted to generate initial annotations for help-need and help-content labels. After deduplication, the model was additionally prompted to generate three plausible but incorrect options for each instance for multiple-choice question evaluation. All annotations and distractors were then reviewed by the authors, resulting in 1K validated instances, with 78.89% of the original data retained. For 12.5% of the retained instances, the segment's start or end time was adjusted to exclude explicit visual help signals (e.g., user turning to Google Search) to ensure fair evaluation. Overall, 66% of the instances were labeled as help-needed, while the remaining 34% required no help.

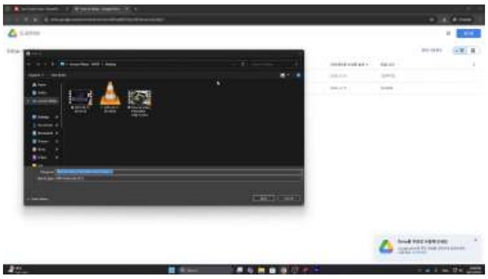
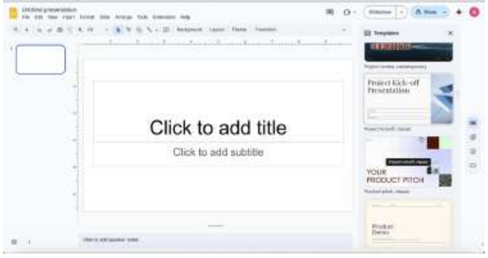
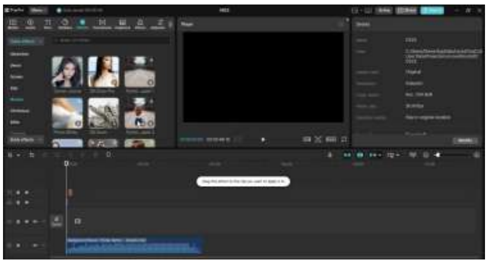
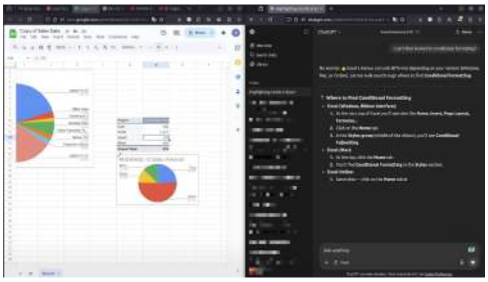
Screenshot	User Behavior State
 *"Okay, I downloaded it already. Delete my test, so I don't get confused. I have the video."*	**Software**: Premiere Pro **Task**: Edit a short instructional video to clearly guide a process. **Behavior State**: **Task Understanding and Preparation** The user is preparing their digital workspace before starting the editing task. They locate the necessary video file on their desktop and delete a superfluous 'test' file to prevent confusion.
 *"I would like to just use this design or the white some minimalistic like iOS design. Oh, this one. This one looks good. Okay, let's just..."*	**Software**: Google Slides **Task**: Create a product pitch deck highlighting a product's key features. **Behavior State**: **Exploration and Decision-Making** The user is actively browsing and comparing different templates, as shown by the scrolling and hovering behavior. The narration ('This one looks good') confirms they are evaluating options to make a final decision.
 *"Okay, that's strange. That's very strange, honestly."*	**Software**: CapCut **Task**: Design a creative intro using animated text. **Behavior State**: **Frustration** The user verbally expresses confusion ('that's strange') after the software behaved in an unexpected way. They are momentarily paused, indicating a blocker in their workflow before they decide on a new course of action.
 *(no narration)*	**Software**: Google Sheets **Task**: Summarize and visualize product sales by category or region. **Behavior State**: **Seeking External Help** The user is unable to find a feature and turns to ChatGPT for assistance. They type a question clarifying their problem, wait for the response, and then read the provided instructions.

Table 6.2: Example instances for the (1) User Behavior State Detection task.

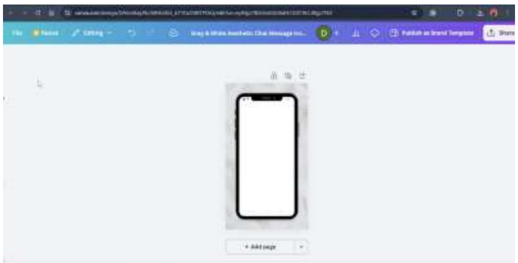
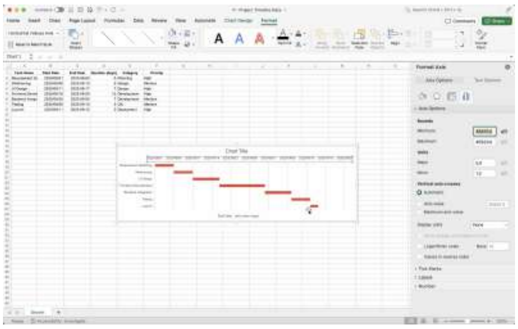
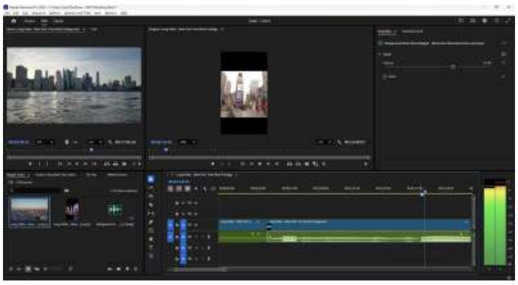
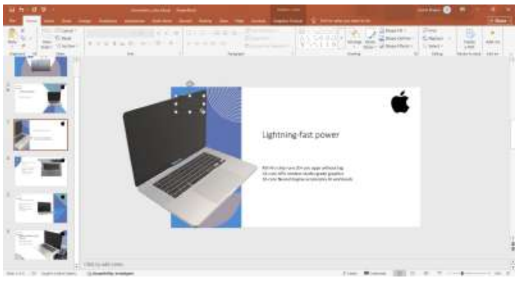
Screenshot	Intent
 *“So now that I have the frame as a design base, I need to include the input field for name, email.”*	**Software**: Canva **Task**: Design a mobile sign-up screen for a fictional app. **Intent**: A: Rename the design file to reflect the new project **B: Add the required input fields to the design** C: Search for a suitable illustration to use as a header D: Resize the canvas to a custom dimension
 *“okay looks perfect, I need to adjust the end date as well”*	**Software**: Excel **Task**: Design a Gantt chart for a mini project. **Intent**: **A: Adjust the end date of the chart’s horizontal axis** B: Adjust the date interval of the chart’s horizontal axis C: Reverse the order of the chart’s vertical axis D: Adjust the start date of the chart’s horizontal axis
 *“When this slot comes, we should put some kind of image here.”*	**Software**: Premiere Pro **Task**: Transform a long video into a short-form clip. **Intent**: A: Create a new text layer above the existing video track **B: Add an image to a specific empty slot in the timeline** C: Apply a transition effect to the end of a video clip D: Add a video clip to the end of the current sequence
 *“Paste, paste, paste, paste. Done. Done.”*	**Software**: PowerPoint **Task**: Create a product pitch deck highlighting a product’s key features. **Intent**: A: Align the logos with the main text boxes. B: Delete the logos from all the slides. **C: Duplicate the logos onto the remaining slides.** D: Change the color of the logos on all slides.

Table 6.3: Example instances for the (2) Intent Prediction task.

6.3 Experiments

6.3.1 Experimental Setup

We evaluate a range of multimodal large language models (MLLMs) on our benchmark to assess their ability to understand, reason about, and assist users in open-ended software workflows. Our evaluation includes eight representative MLLMs spanning both proprietary and open-source models: **Gemini-2.5-Flash** [62], **Gemini-2.5-Pro** [62], **GPT-4o-mini** [136], **GPT-4o** [136], **Claude-4.5-Sonnet** [6], **Qwen3-VL-8B** [162], **InternVideo2.5-Chat-8B** [172], and **InternVL3-8B** [207]. All models are evaluated in a zero-shot setting using publicly available APIs or checkpoints, without any additional fine-tuning.

For each test instance, we uniformly sample 32 frames from the corresponding video segment, providing only visual input (excluding narration audio) to simulate perception based solely on visual cues. To ensure consistency across models, we use standardized prompting templates. We also prompt models to generate both a predicted label and supporting reasoning, a strategy shown to improve task performance [87].

Our main experiments are conducted in an offline inference setting, where models are given the full video segment to solve the task. To approximate real-world proactive assistant scenarios, we additionally evaluate an online setting, in which the model receives visual input progressively. Specifically, at 25%, 50%, 75%, and 100% of the segment, we uniformly sample 32 frames from the corresponding prefix for inference.

6.3.2 Evaluation Tasks

(1) Behavior State Detection. This task measures whether a model can identify the user's behavioral state from a given video segment. We provide each model with clips and ask it to classify them into one of nine taxonomy-defined states. Two configurations are tested: (*i*) using only the current segment and (*ii*) with prior history, where the model is given the immediately preceding segment's behavior state. This is framed as a multi-class classification problem, and performance is evaluated using accuracy.

$$\text{Accuracy} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}(\hat{y}_i = y_i) \tag{6.1}$$

(2) Intent Prediction. This task evaluates a model's ability to infer the user's underlying goal within a given video segment. Models are prompted to predict what the user is trying to accomplish in two settings: (*i*) using only the current segment, and (*ii*) with additional context from the detected behavior state, where the model is also given the state label and its definition. We adopt a multiple-choice question (MCQ) format, where the model selects the most likely intent from four candidate options. Performance is measured using accuracy. For the default setting (*i*), we additionally report multi-binary accuracy (MBAcc) following prior work [21, 36, 31], which evaluates whether the model correctly identifies the ground-truth intent in all three pairwise comparisons against incorrect alternatives.

Accuracy. Measures the proportion of instances where the model selects the correct intent option from the four candidates.

$$\text{Accuracy} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}(\hat{y}_i = y_i) \tag{6.2}$$

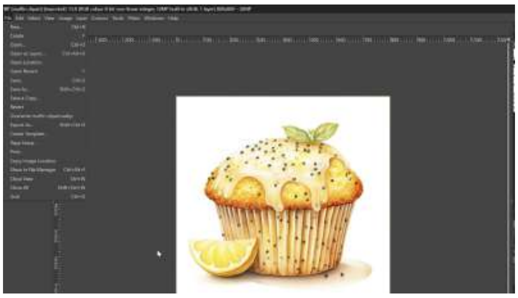
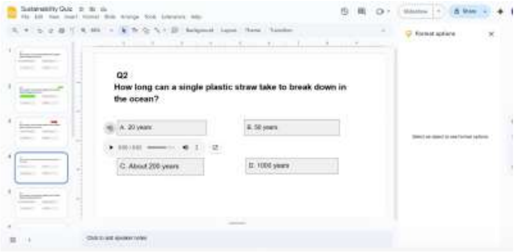
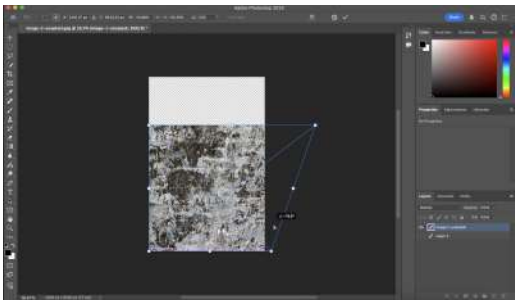
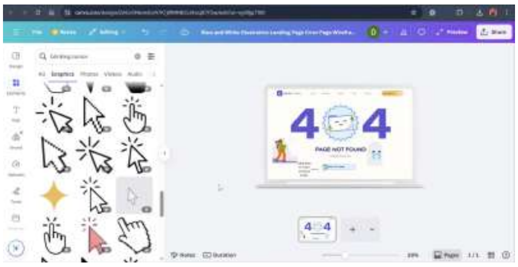
Screenshot	Help
 *"Where could I insert the text? [...] I'm just going to, because the help function I don't quite understand, but I can see if I can add it. Find it in Google."*	**Software**: GIMP **Task**: Create a bakery logo with a warm, friendly identity. **Help Content**: A: how to add another image as a layer **B: find the tool to add text** C: remove the image background D: add a background color or shape
 Q2 How long can a single plastic straw take to break down in the ocean? A. 20 years B. 50 years C. About 200 years D. 1000 years *"I think I made a mistake here and I need to rectify this."*	**Software**: Google Slides **Task**: Create a quiz deck with multiple-choice questions testing sustainability facts **Help Content**: A: align the answer choice boxes B: how to create a quiz slide template **C: how to fix a self-identified audio related error** D: add animation to reveal the correct answer
 *"I'll scale it. I just want to scale this up. How do I keep it?"*	**Software**: Photoshop **Task**: Create a composite from two images. **Help Content**: A: how to use the perspective or warp transform tools B: center the new layer on the canvas C: how to use layer blend modes **D: maintain aspect ratio while scaling**
 404 PAGE NOT FOUND *"So I believe this is, this is great. I believe it's just simple."*	**Software**: Canva **Task**: Design a custom 404 error page with a visual and animated element. **Help Need**: A: help needed **B: no help needed**

Table 6.4: Example instances for the (3) Help Prediction task. For the Help Need Detection task, the top three instances are labeled as *help needed*.

Multi-Binary Accuracy (MBAcc). Following prior work [21, 36], we employ MBAcc to evaluate robustness against distractors. For a given sample i, let y_i be the correct option and $\mathcal{C}_i^- = \{c_{i,1}, c_{i,2}, c_{i,3}\}$ be the set of three incorrect distractor options. The model performs a pairwise comparison function $f(x, \mathrm{opt}_A, \mathrm{opt}_B)$ which returns the chosen option between A and B. A prediction is considered correct under MBAcc only if the model prefers the ground truth y_i over *every* distractor in $\mathcal{C}_i^-$.

$$\mathrm{MBAcc} = \frac{1}{N} \sum_{i=1}^{N} \left(\prod_{c \in \mathcal{C}_i^-} \mathbb{I}(f(x_i, y_i, c) = y_i) \right) \tag{6.3}$$

(3) Help Prediction. The final task evaluates whether models can move beyond understanding and reasoning to provide actionable assistance. Given a video segment, models are asked to predict whether the user requires help (*Need*), and if so, what kind of help would be most appropriate (*Content*). **Help Need Detection** is framed as a binary classification task and evaluated using accuracy, precision, recall, and F1-score. **Help Content Prediction**, similar to Intent Prediction, uses a multiple-choice question (MCQ) format and is evaluated using accuracy and multi-binary accuracy (MBAcc) for the default setting. We test three settings for both tasks: (*i*) video only, (*ii*) video + behavior state, where the model is given the behavior label and its definition for the current segment, and (*iii*) video + behavior state + intent, where the model additionally receives the identified user intention. These settings progressively assess the model's ability to leverage layered user context for meaningful, situation-aware assistance.

Help Need Detection

- **Accuracy**: The ratio of correctly predicted observations to total observations.

$$\mathrm{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{6.4}$$

- **Precision**: The ratio of correctly predicted positive observations to the total predicted positives.

$$\mathrm{Precision} = \frac{TP}{TP + FP} \tag{6.5}$$

- **Recall**: The ratio of correctly predicted positive observations to the all observations in the actual class.

$$\mathrm{Recall} = \frac{TP}{TP + FN} \tag{6.6}$$

- **F1-Score**: The harmonic mean of Precision and Recall.

$$\mathrm{F1} = 2 \cdot \frac{\mathrm{Precision} \cdot \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}} \tag{6.7}$$

Help Content Prediction Same as the metrics used in **(2) Intent Prediction**.

6.3.3 Results

Table 6.5 presents the performance of baseline models on GUIDE across the tasks, with accuracies reported under default and context-augmented settings. Overall, models performed weakest on Behavior State Detection and Help Prediction, with default-setting accuracies peaking at 44.61% and 55.00% for Behavior State Detection and Help Content Prediction, respectively, both from Claude-4.5-Sonnet [6]. While Gemini-2.5-Pro [62] reached nearly 70% accuracy on Help Need Detection, most other models showed substantially lower performance across both Help sub-tasks. Across tasks, we observe that models generally benefit from added behavioral and intent context, with particularly notable improvements in help-related predictions. We report the main findings below.

Model	**(1) Behavior Detection**		**(2) Intent Prediction**		**(3) Help Prediction**					
	–	*+ Prev.*	–	*+ Behavior*	**Help Need Detection**			**Help Content Prediction**		
					–	*+Bhv.*	*+Bhv.+Itnt.*	–	*+Bhv.*	*+Bhv.+Itnt.*
Gemini-2.5-Flash [62]	36.91	38.19	65.40	66.77	53.64	76.33	78.07	49.53	53.75	78.59
Gemini-2.5-Pro [62]	42.44	43.79	67.80	70.16	**69.82**	84.73	82.38	52.74	57.03	79.69
GPT-4o-mini [136]	17.65	17.07	60.76	62.19	46.05	78.92	82.26	31.32	42.86	79.84
GPT-4o [136]	36.32	37.24	61.19	62.58	49.69	**87.79**	**87.91**	45.95	48.37	79.78
Claude-4.5-Sonnet [6]	**44.61**	**45.63**	**71.39**	**72.62**	39.49	58.56	59.43	**55.00**	**62.17**	**82.79**
Qwen3-VL-8B [162]	37.97	38.13	62.70	64.03	52.83	70.39	77.36	46.06	50.63	80.11
InternVideo2.5-8B [172]	21.57	27.02	43.79	45.13	34.36	35.35	35.25	23.67	29.15	73.86
InternVL3-8B [207]	22.57	24.90	46.11	46.97	34.94	43.73	46.82	27.03	32.20	72.97

Table 6.5: Evaluation results on accuracy across (1) Behavior State Detection, (2) Intent Prediction, and (3) Help Prediction.

Behavior State Detection

Behavior state detection remains highly challenging. All models struggled to accurately infer the user's behavioral state from video segments, underscoring the difficulty of the 9-way classification task. While proprietary models such as Claude-4.5-Sonnet [6] and Gemini-2.5-Pro [62] performed best, no model surpassed 45% accuracy, and most fell below 40%.

Models often misinterpret signals of struggle. The most common failure was misclassifying *Frustration* or *Debugging* as *Performing Actions* or *Exploration and Decision-Making* (Figure 6.5). These errors reveal a critical limitation in current MLLMs: a systemic bias toward interpreting interactions as productive execution while failing to recognize signs of struggle or hesitation. While models achieve reasonable accuracy for visually distinct states like *Seeking External Help* (0.61) and *Performing Actions* (0.57), they show near-zero capability in detecting *Frustration* (0.07) and *Debugging* (0.04). Instead, these negative states are overwhelmingly misclassified as *Performing Actions* (39% and 43%, respectively) or *Exploration and Decision-Making* (31% and 29%). This suggests that models perceive the visual activity of a struggling user—such as repeated clicking or rapid mouse movements—as deliberate progress, lacking the temporal understanding to distinguish between trial-and-error and confident execution.

Temporal context shows modest potential. Incorporating the prior behavior state led to small but consistent gains across models. While most improvements were marginal, the largest gain was observed for InternVideo2.5-8B [172] with 5.45 percentage points, suggesting that temporal context holds value and may be more effectively utilized with improved temporal reasoning capabilities.

Intent Prediction

Intent prediction is the most tractable task, but still imperfect. Among the three tasks, models achieved the highest performance on intent prediction, with several surpassing 60% accuracy. However, performance drops under the stricter MBAcc metric, which requires consistent discrimination across all answer pairs. This indicates that while models can often select a plausible intent, they still struggle with

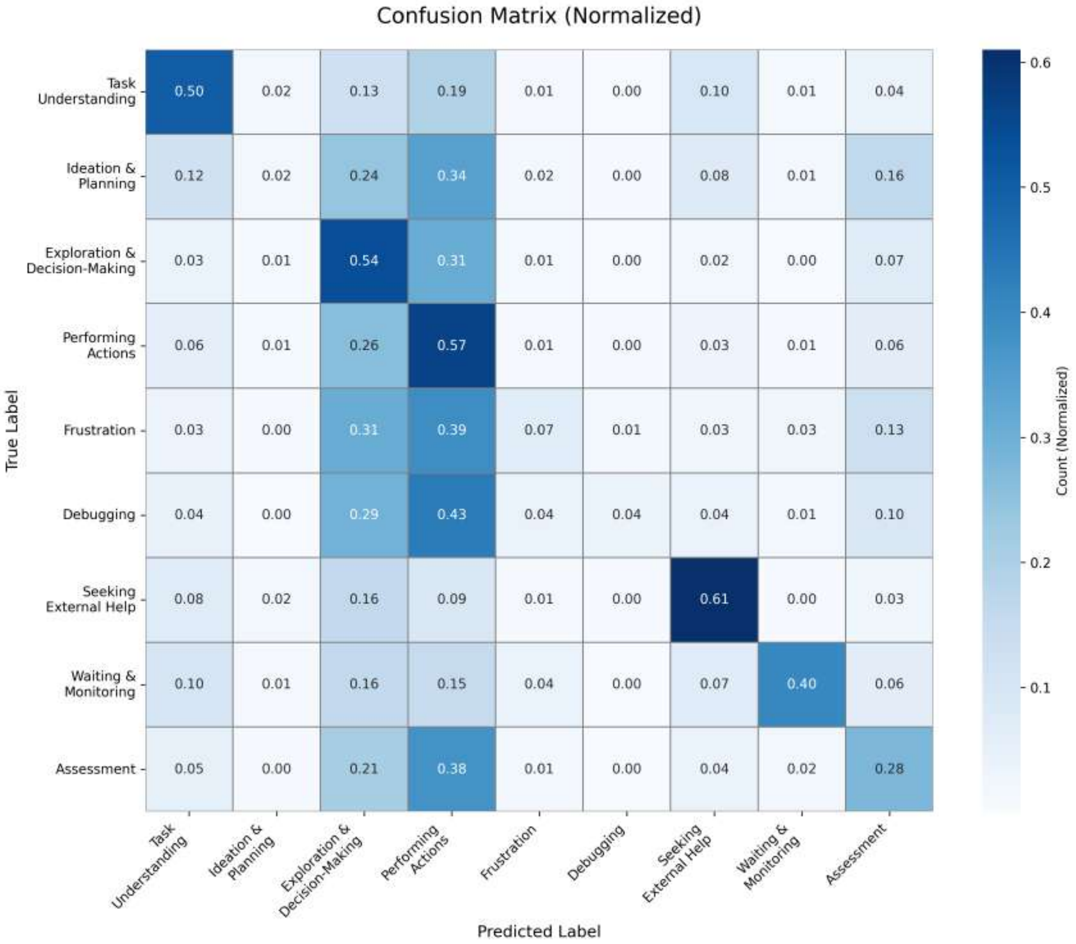


Figure 6.5: Normalized confusion matrix for user behavior state classification. The most common errors occur when *Frustration* or *Debugging* is misclassified as *Performing Actions* or *Exploration and Decision-Making.*

reliably identifying the correct one over all distractors (Table 6.7).

Behavior context helps, but only slightly. Incorporating behavior state context (i.e., the user's behavioral label and definition) consistently improved performance, but the gains were relatively modest across all models. This suggests that while such context may offer useful cues, it does not provide sufficient information on its own or is not yet effectively leveraged by current models for intent inference.

Help Prediction

High Variance and Missed Help Cases in Need Detection. Table 6.6 shows the full performance results for Help Need Detection. This subtask exhibited the most variance across models, with F1 scores ranging from 0.31 (InternVideo2.5-8B [172]) to 77.42 (Gemini-2.5-Pro [62]). Notably, recall was particularly low across most models—except for Gemini-2.5-Pro, all others had recall under 37. This indicates that many instances where users actually needed help were misclassified as not needing it, echoing similar trends in Behavior State Detection (Section 6.3.3) where models frequently misinterpreted signals of struggle.

Model	Help Need Detection											
	–				*+ Behavior State*				*+ Behavior State + Intent*			
	Acc	*Prec*	*Rec*	*F1*	*Acc*	*Prec*	*Rec*	*F1*	*Acc*	*Prec*	*Rec*	*F1*
Gemini-2.5-Flash [62]	53.64	83.27	36.62	50.87	76.33	97.67	65.47	78.39	78.07	94.56	70.62	80.86
Gemini-2.5-Pro [62]	**69.82**	76.42	**78.09**	**77.42**	84.73	93.61	82.34	87.61	82.38	91.20	80.94	86.76
GPT-4o-mini [136]	46.05	83.03	22.31	35.17	76.73	97.61	66.23	78.92	79.71	97.20	71.29	82.26
GPT-4o [136]	49.69	74.41	35.14	47.73	**87.79**	95.39	**85.53**	**90.19**	**87.91**	95.12	**85.95**	**90.30**
Claude-4.5-Sonnet [6]	39.49	**87.69**	8.92	16.19	58.56	**99.16**	37.09	53.99	59.43	**99.19**	38.44	55.41
Qwen3-VL-8B [162]	52.83	79.86	34.23	47.92	70.39	94.35	58.50	72.22	77.36	95.38	67.56	79.09
InternVideo2.5-8B [172]	34.36	33.33	0.16	0.31	35.35	90.91	1.56	3.07	35.25	83.33	1.56	3.07
InternVL3-8B [207]	34.94	72.73	1.25	2.46	43.73	98.88	15.77	27.20	46.82	98.40	19.22	32.16

Table 6.6: Results for **Help Need Detection** on accuracy, precision, recall, and F1-score across three conditions (default, with behavior state, with behavior state and intent).

Behavior state context improves Help Need Detection. Providing the user's behavior state led to consistent and significant improvements in Help Need Detection across all models, with the largest gain observed in GPT-4o [136], which achieved a 42.46-point increase in F1 score. This suggests that context, such as whether a user is exploring, hesitating, or showing signs of frustration, provides strong cues for determining help needs.

Help Content Prediction remains challenging, but benefits from intent context. Help Content Prediction proved particularly challenging, with all models struggling and the top accuracy reaching only 55% from Claude-4.5-Sonnet [6], which further declined to around 50% under the stricter MBAcc evaluation. However, incorporating intent information, representing what the user is trying to accomplish, led to substantial improvements across models. The largest gain was observed in InternVideo2.5-8B [172], with a 50.19 percentage point increase, highlighting the importance of understanding both user state and intent for providing meaningful, targeted support.

Other Findings

Online vs. Offline Setting: models benefit more from temporal Context. In our online simulation experiment, where models are given progressively more of the video segment (25%, 50%, 75%, and 100%), we observe consistent performance gains across all four tasks (Figure 6.6). Gemini-2.5-Flash [62] shows substantial improvements with more visual input, indicating a strong ability to integrate growing context into more accurate predictions. In contrast, InternVideo2.5-8B [172] displays relatively minor gains. These findings suggest that gathering appropriate context over time is crucial for proactive AI assistance, where systems must not only react but also anticipate user needs based on incomplete and evolving information.

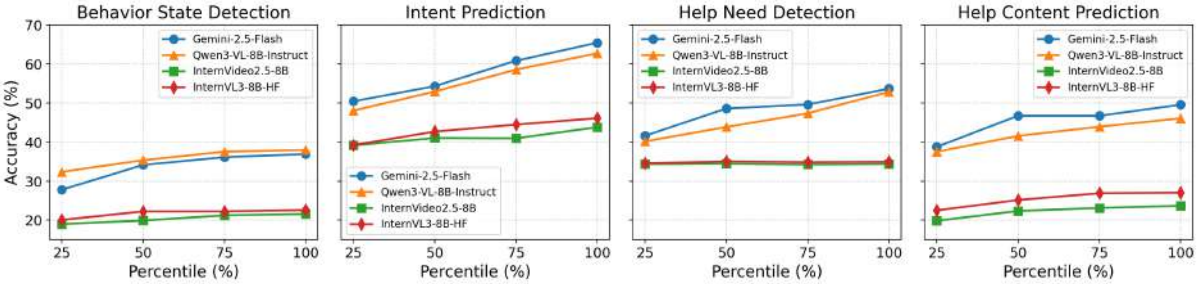


Figure 6.6: Accuracy trends for Gemini-2.5-Flash [62] and InternVideo2.5-8B [172] across the tasks in the online setting, where models are given progressively more of the video segment (25%, 50%, 75%, and 100%). Both models show steady performance gains as they see more segments, while Gemini-2.5-Flash shows larger and more consistent gains.

Model	**Intent Prediction**		**Help Prediction**	
	Acc	*MBAcc*	*Acc*	*MBAcc*
Gemini-2.5-Flash [62]	65.40	59.09	49.53	44.69
Gemini-2.5-Pro [62]	67.80	64.34	52.74	45.31
GPT-4o-mini [136]	60.76	50.24	31.32	28.59
GPT-4o [136]	61.19	56.58	45.95	41.25
Claude-4.5-Sonnet [6]	**71.39**	**65.44**	**55.00**	**50.78**
Qwen3-VL-8B [162]	62.70	58.07	46.06	44.69
InternVideo2.5-8B [172]	43.79	27.98	23.67	18.75
InternVL3-8B [207]	46.11	40.75	27.03	23.75

Table 6.7: Evaluation of **Intent Prediction** and **Help Content Prediction**, with Accuracy (Acc) and Multi-Binary Accuracy (MBAcc).

6.4 Conclusion

We introduced a benchmark for evaluating models in understanding, reasoning about, and assisting users in open-ended GUI-based workflows. Grounded in real novice user demonstrations, our tasks—behavior state detection, intent prediction, and help prediction—capture core capabilities needed for collaborative GUI Agents. Evaluation across state-of-the-art MLLMs revealed that models struggle to interpret nuanced user behavior and accurately infer assistance needed in open-ended GUI scenarios. However, when provided with appropriate user context, such as behavior state and intent, models showed consistent improvements, highlighting the value of structured user understanding in enhancing model support capabilities. Unlike prior benchmarks that primarily focus on action recognition, our work emphasizes user cues related to cognition, behavior, and intent that agents must interpret to collaborate effectively with people. Overall, our benchmark lays the groundwork for developing user-aware agents that support human workflows.

Chapter 7. Discussion

This dissertation set out to bridge the gap between the linear, unstructured nature of procedural video and the dynamic, non-linear needs of learners. Through the development of structural frameworks (VideoMix, Beyond Instructions) and assistance systems (SoftVideo, GUIDE), I have demonstrated that augmenting video with **Contextual Units**—semantic structures that define the *what*, *how*, and *why* of a procedure—can effectively scaffold the full Video Learning Cycle.

In this chapter, I synthesize findings across the four projects to discuss the broader implications of this work. I first examine how the granularity of contextual units should adapt to the user's learning phase (Section 7.1). I then discuss design principles for selecting effective contextual units (Section 7.2). Next, I outline directions for adaptive procedural support through user modeling (Section 7.3). Finally, I discuss the generalizability of contextual units (Section 7.4).

7.1 The Dynamic Nature of Contextual Units

A central finding of this thesis is that there is no single "atomic unit" of procedural knowledge. Traditional approaches often rely on static temporal segmentation, such as dividing a video into chronological steps or chapters. However, the projects in this dissertation show that the most useful unit of analysis is not fixed. It shifts with the user's goals and phase within the learning lifecycle. As users move from exploring a task to executing it, the type of contextual unit that supports their progress changes accordingly.

In the **Exploration phase**, users benefit from macro-level units that help them understand what the procedure consists of and how different tutorials compare. VideoMix [190] showed that users reason at the level of Outcomes, Approaches, and Methods when forming a mental model of the task landscape. At this stage, the unit is essentially the "what" of a procedure. High-level structure helps users decide which strategy fits their needs.

In the **Comprehension phase**, once users commit to a tutorial, the relevant unit shifts to more fine-grained semantic content. Beyond Instructions [188] revealed that users attend not only to instructions, but also to other information types that clarify the "why" or "how" behind a procedure, such as Justifications, Warnings, or Tool Specifications. These semantic cues help users access and navigate the content efficiently.

In the **Following phase**, the salient unit becomes the "how." As users attempt to carry out the procedure, they need cues that reflect difficulty, effort, and typical pitfalls. SoftVideo [191] demonstrated that interaction-derived signals such as Step Difficulty and Step Relevancy provide meaningful guidance for pacing, identifying struggles, and recovering from errors. These units do not describe the content of the video alone. They capture how people actually experience the procedure, allowing the system to support them as they act.

Finally, in the **Assistance** phase, the contextual unit shifts again toward modeling user cognition. GUIDE showed that intelligent assistance depends on understanding both *why* the user is acting (Intent) and *how* they are progressing (Behavior State), such as moments of Frustration. They allow assistive agents to decide when to intervene and what support to offer.

Across the four projects, I demonstrate that contextual units are dynamic and phase-specific. This

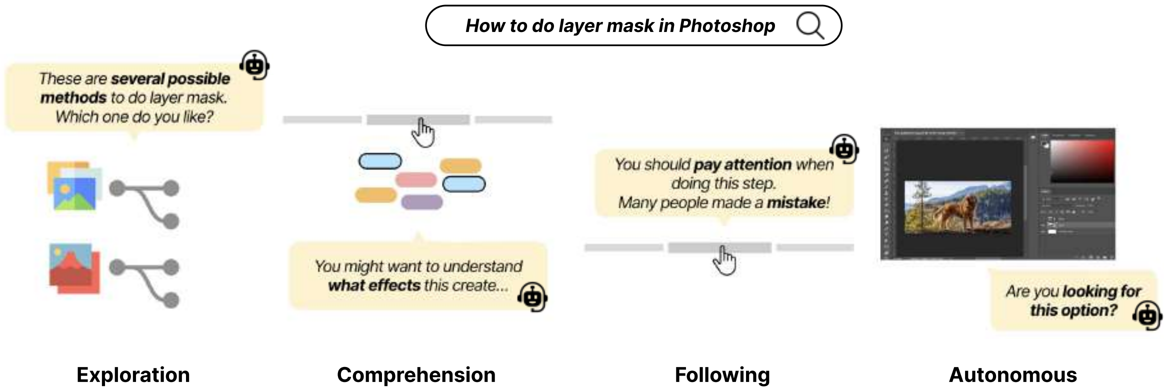

How to do layer mask in Photoshop
These are several possible methods to do layer mask. Which one do you like?
You might want to understand what effects this create...
You should pay attention when doing this step. Many people made a mistake!
Are you looking for this option?
Exploration
Comprehension
Following
Autonomous

Figure 7.1: The same user query can be supported differently depending on the learning phase, with systems adapting contextual units to the user's current goal.

progression suggests that future procedural learning interfaces should adapt their level of granularity to the user's current phase. Although I presented the task learning cycle linearly as Exploration, Comprehension, Following, and Autonomous, task learning is inherently iterative. Users move back and forth between phases rather than progressing in a fixed order. In the GUIDE dataset, I observed many instances of users transitioning from the Autonomous phase back to earlier phases, such as opening a tutorial video alongside the software to follow or replicate specific steps. These observations show that any learning phase can occur at any point in a user's workflow. As a result, effective user support must be conditioned on an ongoing inference of the user's current learning phase, rather than assuming a fixed progression. Adapting to these shifts is key to supporting real procedural learning in practice.

7.2 Design Principles for Selecting Effective Contextual Units

Based on the findings, I present design principles for selecting contextual units that support task learning. First, contextual units should be selected according to the learner's current goal, which is closely tied to their learning phase. When users seek to understand an underlying concept, they benefit from informational units such as justifications, which are most relevant during the Comprehension phase. In contrast, when users attempt to correct an error or replicate an action, they benefit more from interaction-derived units, such as relevant steps demonstrated by other users, which align with the Following phase.

This distinction has direct implications for system responses. For example, when a user asks, "How do I create a layer mask in Photoshop?", current systems typically provide a single, static answer. However, the appropriate response depends on the user's underlying goal (Figure 7.1). If the user's goal is to explore available options, the system should present multiple possible approaches. If the goal is to understand the underlying concept, the system should explain what a layer mask does and why it is useful. When the goal is to execute or replicate an action, the system can highlight common pitfalls or critical steps during execution. Finally, when the user's goal is to work independently but resolve emerging issues, the system should provide targeted, proactive assistance when difficulty is detected.

Furthermore, contextual units should be chosen to reduce the dominant cognitive burden at the user's current moment. During the Exploration phase, the primary challenge lies in searching and comparing alternatives, so units should focus on reducing discovery and comparison costs. During the Comprehension phase, the challenge shifts to extracting relevant segments from long or dense instructional content, making concise explanatory units particularly useful.

Taken together, these principles emphasize that effective assistance in task learning depends not on static content delivery, but on dynamically aligning contextual units with the learner's evolving goals, phase, and cognitive demands. By selecting units that reflect where users are and what they need at a given moment, systems can provide support that is timely, relevant, and aligned with real learning behavior.

7.3 User Modeling for Adaptive Procedural Support

While contextual units such as approaches, information types, and step-level cues can broadly benefit learners, their effectiveness can be significantly enhanced when grounded in user modeling. Capturing user-specific context allows systems to select and present contextual units in ways that better align with individual needs and learning trajectories, rather than offering uniform, static support.

Learners differ in their prior knowledge, preferences, and behavioral patterns, all of which shape how they interpret instructions and where they may require guidance. Signals such as navigation behavior, software interaction logs, and other engagement patterns can reveal these differences in a non-intrusive manner. For example, how a user typically interacts with software when confident versus when struggling can help a system infer their current progress and determine whether support is needed. By leveraging such signals, systems can move from static presentation to dynamic recommendation, such as prioritizing information types a user is likely to value based on prior experience, or identifying moments of difficulty and offering timely assistance.

Through user modeling, systems can deliver adaptive video experiences that tailor explanations, highlight relevant information, and adjust the level of detail to the learner's current needs. They can also support personalized generation, producing examples or instructions that reflect a user's specific context, such as the tools they are using or missing background knowledge. Over longer periods, such systems can offer longitudinal support by tracking evolving skill profiles, identifying recurring challenges, and adapting assistance as the learner progresses.

These ideas have important implications for the design of AI agents that assist users during task learning. First, effective user-assisting agents must be able to infer which stage of the task-learning cycle a learner is in, so that they can provide appropriate forms of support. Second, they must incorporate user modeling to understand and adapt to diverse factors, including background knowledge, interests, preferences, and habitual workflows. Overall, developing user-aware AI agents opens opportunities for more personalized and adaptive procedural learning. By integrating perception, inference, and interaction, these systems can respect user agency while providing meaningful support across domains, from software-based tasks to hands-on physical activities.

7.4 Generalizability of Contextual Units

The systems and frameworks presented in this dissertation are grounded in specific task domains, which shaped both their design and evaluation. VideoMix and Beyond Instructions focus on tasks with tangible and physical outcomes, such as cooking or assembly. This setting made it possible to clearly demonstrate the value of macro-level contextual units such as outcomes, approaches, and methods. At the same time, this domain focus raises broader questions about how these findings extend to tasks with different learning dynamics or less clearly defined outcomes.

Many of the contextual units introduced in this work are likely to generalize across domains. High-level approaches or alternative methods, for example, are common in a wide range of procedural activities. However, other domains may rely on distinct, domain-specific contextual signals. For example, in music learning, tone, rhythm, or expressive variation may play a central role, while in programming, language-specific constructs, abstractions, or debugging patterns may be more critical. Understanding which contextual units are broadly shared and which are domain-specific remains an important direction for future research.

The latter part of this dissertation focuses on software and GUI-based tasks through SoftVideo and GUIDE, where fine-grained interaction logs and behavioral cues serve as key signals for understanding user state. These domains benefit from rich, readily available interaction data, such as mouse clicks or cursor movements. Extending this approach to non-GUI domains presents new challenges and opportunities. In physical tasks such as cooking or craftwork, cues may instead arise from tool-handling patterns, timing rhythms, or coordination between actions, which may require alternative sensing modalities such as wearables, environmental sensors, or computer vision. Together, these directions suggest that while the specific implementations in this work are domain-bound, the broader concept of contextual units as a link between video content, user behavior, and adaptive support has the potential to generalize across a wide range of procedural learning domains.

Chapter 8. Conclusion and Future Work

This dissertation demonstrated that augmenting procedural videos with granular Contextual Units can effectively support the full lifecycle of human task learning. This chapter summarizes the main contributions of the thesis and proposes future directions.

8.1 Summary of Contributions

- **Video Understanding**: Novel computational pipelines and taxonomies for extracting meaningful semantic structure from unstructured procedural videos.
- **Video Interaction**: Interaction techniques that facilitate the sensemaking of procedural content and bridge the gap between passive viewing and active execution.
- **User-Assisting AI**: Benchmarks and frameworks for developing intelligent agents and assistive interfaces that model high-level user states for building context-aware systems.

8.2 Future Directions

8.2.1 Personalized and Adaptive Videos

Learners differ in their prior knowledge, background, navigation patterns, and responses to difficulty. Through user modeling, future systems could infer a learner profile to tailor recommendations, highlight relevant information, or adjust the level of detail presented in a video. In addition, videos themselves can become "live" instructional materials that evolve based on how learners interact with them. By detecting where users struggle or repeatedly revisit, a system could automatically restructure or annotate videos by inserting clarifying tips, adding pauses, or surfacing alternative explanations at those timestamps. Such personalized and self-evolving tutorials would help learners access the most relevant guidance while continuously improving in response to real usage patterns.

8.2.2 Generative Instructional Media

While this dissertation focuses on structuring and reorganizing existing instructional videos, the contextual units identified here reveal opportunities for generating pedagogically meaningful learning materials. Rather than producing arbitrary content, generative systems could use these units as semantic constraints to create tutorial segments that address specific learner needs. For example, a generative model could be conditioned to amplify scaffolding for novices by synthesizing additional *Justifications* or *Tips*, or conversely, to enhance visual clarity by generating detailed intermediate *Status* of the work. Grounding generative output in structured contextual units would allow future systems to produce instructional media that support more controllable and effective learning.

8.2.3 Collaborative AI Agents for Task Learning

There is a growing opportunity to design collaborative AI agents that assist users while preserving their sense of control. Rather than relying on full automation, future agents could model user intent, anticipate upcoming challenges, and intervene in ways that complement the user's ongoing actions. The form of assistance should flexibly adapt to the user's immediate goals. Users who want to maintain momentum may benefit from explicit help, such as automating a small sub-action or surfacing a highly relevant tutorial segment, while those aiming for deeper procedural understanding may benefit more from implicit help, including gentle cues, clarifications, or highlighting relevant prior interactions. Such scaffolded assistance would help users overcome difficulties without disengaging them from the learning process. Importantly, this perspective also suggests rethinking how we evaluate AI agents for task learning: while many current systems emphasize task completion or success rate, a more meaningful measure is whether the support leads to durable learning, such as whether users can independently resolve similar challenges when they arise again. By grounding interventions in user modeling and evaluating their impact on long-term retention rather than short-term completion, collaborative agents can more effectively support procedural learning.

8.2.4 Longitudinal Support for Skill Development

Future research can also examine procedural learning over longer time horizons. While SoftVideo showed how collective interaction data reveals meaningful patterns within a single tutorial, an equally important direction is modeling a single user across many videos to understand their evolving skill profile. Tracking how learners revisit tutorials, where difficulties persist, and how their reliance on guidance changes over time would enable more nuanced and personalized support. Longitudinal models could recommend different tutorial styles as the learner matures, identify recurring weaknesses across tasks, or gradually adjust the level of assistance to foster greater independence. By combining structured instructional representations with long-term behavioral insights, future systems can support not only immediate task completion but also sustained skill development.

Bibliography

[1] Fusion 360. *Autodesk*, 2022 (accessed February 9, 2022).

[2] Triantafyllos Afouras, Effrosyni Mavroudi, Tushar Nagarajan, Huiyu Wang, and Lorenzo Torresani. HT-step: Aligning instructional articles with how-to videos. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023.

[3] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, and et al. Flamingo: a visual language model for few-shot learning. In *NeurIPS*, 2022.

[4] D.G. Altman. Practical statistics for medical research. 1990.

[5] Lorin W. Anderson, David R. Krathwohl, Peter W. Airasian, Kathleen A. Cruikshank, Richard E. Mayer, Paul R. Pintrich, James Raths, and Merlin C. Wittrock. *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives, Complete Edition*. Longman, 2001.

[6] Anthropic. Introducing claude sonnet 4.5. https://www.anthropic.com/news/claude-sonnet-4-5, 2025. Anthropic News Release, September 29, 2025.

[7] YouTube Data API. *YouTube*, 2022 (accessed February 9, 2022).

[8] Kumar Ashutosh, Zihui Xue, Tushar Nagarajan, and Kristen Grauman. Detours for navigating instructional videos, 2024.

[9] AutoCAD. *Autodesk*, 2022 (accessed February 9, 2022).

[10] Microsoft Azure. *Speech to text*, 2022 (accessed Sep 14, 2022).

[11] Max Bain, Jaesung Huh, Tengda Han, and Andrew Zisserman. Whisperx: Time-accurate speech transcription of long-form audio. *INTERSPEECH 2023*, 2023.

[12] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *ICCV*, 2021.

[13] Nikola Banovic, Tovi Grossman, Justin Matejka, and George Fitzmaurice. Waken: Reverse engineering usage information and interface structure from software videos. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, UIST '12, page 83–92, New York, NY, USA, 2012. Association for Computing Machinery.

[14] Omri Berkovitch, Sapir Caduri, Noam Kahlon, Anatoly Efros, Avi Caciularu, and Ido Dagan. Identifying user goals from ui trajectories. In *Companion Proceedings of the ACM on Web Conference 2025*, WWW '25, page 2381–2390, New York, NY, USA, 2025. Association for Computing Machinery.

[15] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, 2021.

[16] Floraine Berthouzoz, Wilmot Li, and Maneesh Agrawala. Tools for placing cuts and transitions in interview video. *ACM Transactions on Graphics*, 31(4), jul 2012.

[17] Benjamin S. Bloom. *Taxonomy of educational objectives: The classification of educational goals.* Longman Group, 1st edition, 1956.

[18] Christopher Brinton, Swapna Buccapatnam, Mung Chiang, and H. Vincent Poor. Mining mooc clickstreams: On the relationship between learner video-watching behavior and in-video quiz performance. volume 64, pages 1–1, 07 2016.

[19] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

[20] Peter Brusilovsky. Methods and techniques of adaptive hypermedia. In *Adaptive hypertext and hypermedia*, pages 1–43. Springer, 1998.

[21] Mu Cai, Reuben Tan, Jianrui Zhang, Bocheng Zou, Kai Zhang, Feng Yao, Fangrui Zhu, Jing Gu, Yiwu Zhong, Yuzhang Shang, Yao Dou, Jaden Park, Jianfeng Gao, Yong Jae Lee, and Jianwei Yang. Temporalbench: Towards fine-grained temporal understanding for multimodal video models. *arXiv preprint arXiv:2410.10818*, 2024.

[22] Joao Carreira and Andrew Zisserman. Action recognition? a new model and the kinetics dataset. In *CVPR*, 2017.

[23] Minsuk Chang, Leonore V. Guillain, Hyeungshik Jung, Vivian M. Hare, Juho Kim, and Maneesh Agrawala. Recipescape: An interactive tool for analyzing cooking instructions at scale. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, page 1–12, New York, NY, USA, 2018. Association for Computing Machinery.

[24] Minsuk Chang, Mina Huh, and Juho Kim. Rubyslippers: Supporting content-based voice navigation for how-to videos. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, New York, NY, USA, 2021. Association for Computing Machinery.

[25] Minsuk Chang, Ben Lafreniere, Juho Kim, George Fitzmaurice, and Tovi Grossman. Workflow graphs: A computational model of collective task strategies for 3d design software. In *Proceedings of Graphics Interface 2020*, GI 2020, pages 114 – 124. Canadian Human-Computer Communications Society / Socíeté canadienne du dialogue humain-machie, 2020.

[26] Minsuk Chang, Anh Truong, Oliver Wang, Maneesh Agrawala, and Juho Kim. How to design voice based navigation for how-to videos. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–11, New York, NY, USA, 2019. Association for Computing Machinery.

[27] Hsiang-Ting Chen, Li-Yi Wei, Björn Hartmann, and Maneesh Agrawala. Data-driven adaptive history for image editing. I3D '16, page 103–111, New York, NY, USA, 2016. Association for Computing Machinery.

[28] Valerie Chen, Alan Zhu, Sebastian Zhao, Hussein Mozannar, David Sontag, and Ameet Talwalkar. Need help? designing proactive ai assistants for programming. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, CHI '25, New York, NY, USA, 2025. Association for Computing Machinery.

[29] Yuexi Chen, Vlad I Morariu, Anh Truong, and Zhicheng Liu. Tutoai: a cross-domain framework for ai-assisted mixed-media tutorial creation on physical tasks. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, CHI '24, New York, NY, USA, 2024. Association for Computing Machinery.

[30] Feng Cheng and Gedas Bertasius. Tallformer: Temporal action localization with long-memory transformer. In *ECCV*, 2022.

[31] Jen-Hao Cheng, Vivian Wang, Huayu Wang, Huapeng Zhou, Yi-Hao Peng, Hou-I Liu, Hsiang-Wei Huang, Kuang-Ming Chen, Cheng-Yen Yang, Wenhao Chai, et al. Tempura: Temporal event masked prediction and understanding for reasoning in action. *arXiv preprint arXiv:2505.01583*, 2025.

[32] Peggy Chi, Nathan Frey, Katrina Panovich, and Irfan Essa. Automatic instructional video creation from a markdown-formatted tutorial. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, UIST '21, page 677–690, New York, NY, USA, 2021. Association for Computing Machinery.

[33] Pei-Yu Chi, Sally Ahn, Amanda Ren, Mira Dontcheva, Wilmot Li, and Björn Hartmann. Mixt: Automatic generation of step-by-step mixed media tutorials. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, UIST '12, page 93–102, New York, NY, USA, 2012. Association for Computing Machinery.

[34] Pei-Yu Chi, Sally Ahn, Amanda Ren, Mira Dontcheva, Wilmot Li, and Björn Hartmann. Mixt: Automatic generation of step-by-step mixed media tutorials. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, UIST '12, page 93–102, New York, NY, USA, 2012. Association for Computing Machinery.

[35] Pei-Yu Chi, Joyce Liu, Jason Linder, Mira Dontcheva, Wilmot Li, and Bjoern Hartmann. Democut: Generating concise instructional videos for physical demonstrations. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, UIST '13, page 141–150, New York, NY, USA, 2013. Association for Computing Machinery.

[36] Jang Hyun Cho, Andrea Madotto, Effrosyni Mavroudi, Triantafyllos Afouras, Tushar Nagarajan, Muhammad Maaz, Yale Song, Tengyu Ma, Shuming Hu, Hanoona Rasheed, Peize Sun, Po-Yao Huang, Daniel Bolya, Suyog Jain, Miguel Martin, Huiyu Wang, Nikhila Ravi, Shashank Jain, Temmy Stark, Shane Moon, Babak Damavandi, Vivian Lee, Andrew Westbury, Salman Khan, Philipp Krähenbühl, Piotr Dollár, Lorenzo Torresani, Kristen Grauman, and Christoph Feichtenhofer. Perceptionlm: Open-access data and models for detailed visual understanding. *arXiv:2504.13180*, 2025.

[37] Bogeum Choi, Sarah Casteel, Jaime Arguello, and Robert Capra. Better understanding procedural search tasks: Perceptions, behaviors, and challenges. *ACM Trans. Inf. Syst.*, 42(3), December 2023.

[38] Jinhan Choi, Jeongyun Han, Woochang Hyun, Hyunchul Lim, Sun Young Huh, SoHyun Park, and Bongwon Suh. Leveraging smartwatches to estimate students' perceived difficulty and interest in online video lectures. In *Proceedings of the 2019 11th International Conference on Education Technology and Computers*, pages 171–175, 2019.

[39] Konstantinos Chorianopoulos. Collective intelligence within web video. *Human-centric Computing and Information Sciences*, 3(1):1–16, 2013.

[40] Anthony Colas, Seokhwan Kim, Franck Dernoncourt, Siddhesh Gupte, Daisy Zhe Wang, and Doo Soon Kim. Tutorialvqa: Question answering dataset for tutorial videos. In *International Conference on Language Resources and Evaluation*, 2019.

[41] Firebase Realtime Database. *Firebase*, 2022 (accessed February 9, 2022).

[42] Maureen Daum, Enhao Zhang, Dong He, Magdalena Balazinska, Brandon Haynes, Ranjay Krishna, Apryle Craig, and Aaron Wirsingn. Vocal: Video organization and interactive compositional analytics. In *The Conference on Innovative Data Systems Research (CIDR)*, 2022.

[43] Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Samuel Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2web: Towards a generalist agent for the web, 2023.

[44] Jonathan D. Denning, William B. Kerr, and Fabio Pellacini. Meshflow: Interactive visualization of mesh construction sequences. volume 30. Association for Computing Machinery, New York, NY, USA, jul 2011.

[45] Descript. Descript, 2022 (accessed Sep 6, 2022).

[46] Himel Dev and Zhicheng Liu. Identifying frequent user tasks from application logs. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces*, IUI '17, page 263–273, New York, NY, USA, 2017. Association for Computing Machinery.

[47] H. Doughty, I. Laptev, W. Mayol-Cuevas, and D. Damen. Action modifiers: Learning from adverbs in instructional videos. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 865–875, Los Alamitos, CA, USA, jun 2020. IEEE Computer Society.

[48] Hazel Doughty and Cees G. M. Snoek. How Do You Do It? Fine-Grained Action Understanding with Pseudo-Adverbs. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[49] W. Keith Edwards, Takeo Igarashi, Anthony LaMarca, and Elizabeth D. Mynatt. A temporal model for multi-level undo and redo. In *Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology*, UIST '00, page 31–40, New York, NY, USA, 2000. Association for Computing Machinery.

[50] José Miguel Santos Espino. Anatomy of instructional videos: a systematic characterization of the structure of academic instructional videos, 2019.

[51] K. J. Kevin Feng, David W. McDonald, and Amy X. Zhang. Levels of autonomy for AI agents. *Knight First Amendment Institute – AI and Democratic Freedoms Essay Series*, 2025.

[52] Figma. Figma make. https://www.figma.com/make/, 2025. Accessed: 2025-11-14.

[53] Leah Findlater, Karyn Moffatt, Joanna McGrenere, and Jessica Dawson. Ephemeral adaptation: The use of gradual onset to improve menu selection performance. CHI '09, page 1655–1664, New York, NY, USA, 2009. Association for Computing Machinery.

[54] Logan Fiorella and Richard E. Mayer. What works and doesn't work with instructional video. *Comput. Hum. Behav.*, 89(C):465–470, dec 2018.

[55] Raymond Fok, Nedim Lipka, Tong Sun, and Alexa F Siu. Marco: Supporting business document workflows via collection-centric information foraging with large language models. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, CHI '24, New York, NY, USA, 2024. Association for Computing Machinery.

[56] C. Ailie Fraser, Joy O. Kim, Hijung Valentina Shin, Joel Brandt, and Mira Dontcheva. Temporal segmentation of creative live streams. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, page 1–12, New York, NY, USA, 2020. Association for Computing Machinery.

[57] C. Ailie Fraser, Julia M. Markel, N. James Basa, Mira Dontcheva, and Scott Klemmer. Remap: Lowering the barrier to help-seeking with multimodal search. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, UIST '20, page 979–986, New York, NY, USA, 2020. Association for Computing Machinery.

[58] C. Ailie Fraser, Tricia J. Ngoon, Mira Dontcheva, and Scott Klemmer. Replay: Contextually presenting learning videos across software applications. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–13, New York, NY, USA, 2019. Association for Computing Machinery.

[59] Daniel Fu, Will Crichton, James Hong, Xinwei Yao, Haotian Zhang, Anh Truong, Avanika Narayan, Maneesh Agrawala, Christopher Ré, and Kayvon Fatahalian. Rekall: Specifying video events using compositions of spatiotemporal labels, 10 2019.

[60] Kiran Gadhave, Jochen Görtler, Zach Cutler, Carolina Nobre, Oliver Deussen, Miriah Meyer, Jeff M. Phillips, and Alexander Lex. Predicting intent behind selections in scatterplot visualizations. *Information Visualization*, 20(4):207–228, 2021.

[61] Difei Gao, Lei Ji, Zechen Bai, Mingyu Ouyang, Peiran Li, Dongxing Mao, Qinchen Wu, Weichen Zhang, Peiyi Wang, Xiangwu Guo, Hengxu Wang, Luowei Zhou, and Mike Zheng Shou. Assistgui: Task-oriented pc graphical user interface automation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13289–13298, Seattle, WA, USA, 2024. IEEE. Benchmarks PC GUI automation with an actor-critic agent framework.

[62] Google Gemini Team. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

[63] Philip J. Guo, Juho Kim, and Rob Rubin. How video production affects student engagement: An empirical study of mooc videos. In *Proceedings of the First ACM Conference on Learning @ Scale Conference*, L@S '14, page 41–50, New York, NY, USA, 2014. Association for Computing Machinery.

[64] Han L. Han, Junhang Yu, Raphael Bournet, Alexandre Ciorascu, Wendy E. Mackay, and Michel Beaudouin-Lafon. Passages: Interacting with text across documents. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, New York, NY, USA, 2022. Association for Computing Machinery.

[65] Tengda Han, Weidi Xie, and Andrew Zisserman. Temporal alignment network for long-term video. In *CVPR*, 2022.

[66] Sandra G Hart and Lowell E Staveland. Development of nasa-tlx (task load index): Results of empirical and theoretical research. *Human mental workload*, 1(3):139–183, 1988.

[67] Version history of Adobe Photoshop. *Adobe*, 2022 (accessed February 9, 2022).

[68] Wenyi Hong, Weihan Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxiao Dong, Ming Ding, and Jie Tang. Cogagent: A visual language model for gui agents, 2023.

[69] Eric Horvitz, Jack Breese, David Heckerman, David Hovel, and Koos Rommelse. The lumière project: Bayesian user modeling for inferring the goals and needs of software users. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, UAI'98, page 256–265, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[70] Bernd Huber, Hijung Valentina Shin, Bryan Russell, Oliver Wang, and Gautham J. Mysore. B-script: Transcript-based b-roll video editing with recommendations. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–11, New York, NY, USA, 2019. Association for Computing Machinery.

[71] Faria Huq, Zora Zhiruo Wang, Frank F. Xu, Tianyue Ou, Shuyan Zhou, Jeffrey P. Bigham, and Graham Neubig. CowPilot: A framework for autonomous and human-agent collaborative web navigation. In Nouha Dziri, Sean (Xiang) Ren, and Shizhe Diao, editors, *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (System Demonstrations)*, pages 163–172, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics.

[72] GNU image manipulation program. *GIMP*, 2022 (accessed February 9, 2022).

[73] Oana Inel, Nava Tintarev, and Lora Aroyo. Eliciting user preferences for personalized explanations for video summaries. In *Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization*, UMAP '20, page 98–106, New York, NY, USA, 2020. Association for Computing Machinery.

[74] Lawrence Jang, Yinheng Li, Charles Ding, Justin Lin, Paul Pu Liang, Dan Zhao, Rogerio Bonatti, and Kazuhito Koishida. Videowebarena: Evaluating long context multimodal agents with video understanding web tasks. 2024.

[75] Yunseok Jang, Yeda Song, Sungryull Sohn, Lajanugen Logeswaran, Tiange Luo, Dong-Ki Kim, Kyunghoon Bae, and Honglak Lee. Scalable Video-to-Dataset Generation for Cross-Platform Mobile Agents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[76] Haojian Jin, Yale Song, and Koji Yatani. Elasticplay: Interactive video summarization with dynamic time budgets. In *Proceedings of the 25th ACM International Conference on Multimedia*, MM '17, page 1164–1172, New York, NY, USA, 2017. Association for Computing Machinery.

[77] Vicky Kalogeiton, Philippe Weinzaepfel, Vittorio Ferrari, and Cordelia Schmid. Action tubelet detector for spatio-temporal action localization. In *ICCV*, 2017.

[78] Anjali Khurana, Xiaotian Su, April Yi Wang, and Parmit K Chilana. Do it for me vs. do it with me: Investigating user perceptions of different paradigms of automation in copilots for feature-rich software. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, CHI '25, New York, NY, USA, 2025. Association for Computing Machinery.

[79] Anjali Khurana, Hariharan Subramonyam, and Parmit K Chilana. Why and when llm-based assistants can go wrong: Investigating the effectiveness of prompt-based interactions for software help-seeking. In *Proceedings of the 29th International Conference on Intelligent User Interfaces*, IUI '24, page 288–303, New York, NY, USA, 2024. Association for Computing Machinery.

[80] Kimia Kiani, George Cui, Andrea Bunt, Joanna McGrenere, and Parmit K. Chilana. Beyond "one-size-fits-all": Understanding the diversity in how software newcomers discover and make use of help resources. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–14, New York, NY, USA, 2019. Association for Computing Machinery.

[81] Jeongyeon Kim, Daeun Choi, Nicole Lee, Matt Beane, and Juho Kim. Surch: Enabling structural search and comparison for surgical videos. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, New York, NY, USA, 2023. Association for Computing Machinery.

[82] Juho Kim, Philip J. Guo, Carrie J. Cai, Shang-Wen (Daniel) Li, Krzysztof Z. Gajos, and Robert C. Miller. Data-driven interaction techniques for improving navigation of educational videos. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST '14, page 563–572, New York, NY, USA, 2014. Association for Computing Machinery.

[83] Juho Kim, Philip J. Guo, Daniel T. Seaton, Piotr Mitros, Krzysztof Z. Gajos, and Robert C. Miller. Understanding in-video dropouts and interaction peaks inonline lecture videos. In *Proceedings of the First ACM Conference on Learning @ Scale Conference*, L@S '14, page 31–40, New York, NY, USA, 2014. Association for Computing Machinery.

[84] Juho Kim, Phu Tran Nguyen, Sarah Weir, Philip J. Guo, Robert C. Miller, and Krzysztof Z. Gajos. Crowdsourcing step-by-step information extraction to enhance existing how-to videos. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, page 4017–4026, New York, NY, USA, 2014. Association for Computing Machinery.

[85] Seoyoung Kim, Arti Thakur, and Juho Kim. *Understanding Users' Perception Towards Automated Personality Detection with Group-Specific Behavioral Data*, page 1–12. Association for Computing Machinery, New York, NY, USA, 2020.

[86] René F. Kizilcec, Chris Piech, and Emily Schneider. Deconstructing disengagement: Analyzing learner subpopulations in massive open online courses. In *Proceedings of the Third International Conference on Learning Analytics and Knowledge*, LAK '13, page 170–179, New York, NY, USA, 2013. Association for Computing Machinery.

[87] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc.

[88] Nicholas Kong, Tovi Grossman, Björn Hartmann, Maneesh Agrawala, and George Fitzmaurice. Delta: a tool for representing and comparing workflows. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, page 1027–1036, New York, NY, USA, 2012. Association for Computing Machinery.

[89] Geza Kovacs. Effects of in-video quizzes on mooc lecture viewing. In *Proceedings of the Third (2016) ACM Conference on Learning @ Scale*, L@S '16, page 31–40, New York, NY, USA, 2016. Association for Computing Machinery.

[90] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *ICCV*, 2017.

[91] Chi-Hsi Kung, Shu-Wei Lu, Yi-Hsuan Tsai, and Yi-Ting Chen. Action-slot: Visual action-centric representations for multi-label atomic activity recognition in traffic scenes. In *CVPR*, 2024.

[92] Benjamin Lafreniere, Tovi Grossman, and George Fitzmaurice. Community enhanced tutorials: improving tutorials with multiple demonstrations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, page 1779–1788, New York, NY, USA, 2013. Association for Computing Machinery.

[93] Minhwa Lee, Zae Myung Kim, Vivek Khetan, and Dongyeop Kang. Human-ai collaborative taxonomy construction: A case study in profession-specific writing assistants. In *Proceedings of the Third Workshop on Intelligent and Interactive Writing Assistants*, In2Writing '24, page 51–57, New York, NY, USA, 2024. Association for Computing Machinery.

[94] Jie Lei, Licheng Yu, Tamara L Berg, and Mohit Bansal. Tvr: A large-scale dataset for video-subtitle moment retrieval. In *ECCV*, 2020.

[95] Kunpeng Li, Chen Fang, Zhaowen Wang, Seokhwan Kim, Hailin Jin, and Yun Fu. Screencast tutorial video understanding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[96] Linjie Li, Yen-Chun Chen, Yu Cheng, Zhe Gan, Licheng Yu, and Jingjing Liu. Hero: Hierarchical encoder for video+language omni-representation pre-training. In *Conference on Empirical Methods in Natural Language Processing*, 2020.

[97] Nan Li, Lukasz Kidzinski, Patrick Jermann, and Pierre Dillenbourg. How do in-video interactions reflect perceived video difficulty? *Proceedings of the European MOOCs Stakeholder Summit 2015*, pages 112–121, 2015.

[98] Nan Li, Łukasz Kidziński, Patrick Jermann, and Pierre Dillenbourg. Mooc video interaction patterns: What do they tell us? In *Design for Teaching and Learning in a Networked World*, pages 197–210. Springer International Publishing, 2015.

[99] Toby Jia-Jun Li, Jingya Chen, Tom M. Mitchell, and Brad A. Myers. Towards effective human-ai collaboration in gui-based interactive task learning agents. In *CHI 2020 Workshop on Artificial Intelligence for HCI: A Modern Approach (AI4HCI)*, 2020.

[100] Jian Liao, Adnan Karim, Shivesh Singh Jadon, Rubaiat Habib Kazi, and Ryo Suzuki. Realitytalk: Real-time speech-driven augmented presentation for ar live storytelling. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*, UIST '22, New York, NY, USA, 2022. Association for Computing Machinery.

[101] David Chuan-En Lin, Fabian Caba Heilbron, Joon-Young Lee, Oliver Wang, and Nikolas Martelaro. Videomap: Supporting video exploration, brainstorming, and prototyping in the latent space. In *Proceedings of the 16th Conference on Creativity & Cognition*, C&C '24, page 311–327, New York, NY, USA, 2024. Association for Computing Machinery.

[102] Georgianna Lin, Jin Yi Li, Afsaneh Fazly, Vladimir Pavlovic, and Khai Truong. Identifying multimodal context awareness requirements for supporting user interaction with procedural videos. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, New York, NY, USA, 2023. Association for Computing Machinery.

[103] Kevin Lin, Linjie Li, Chung-Ching Lin, Faisal Ahmed, Zhe Gan, Zicheng Liu, Yumao Lu, , and Lijuan Wang. Swinbert: End-to-end transformers with sparse attention for video captioning. In *CVPR*, 2022.

[104] Kevin Qinghong Lin, Linjie Li, Difei Gao, Qinchen Wu, Mingyi Yan, Zhengyuan Yang, Lijuan Wang, and Mike Z. Shou. Videogui: A benchmark for gui automation from instructional videos. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024), Datasets and Benchmarks Track*, pages 69329–69360, New Orleans, USA, 2024. Curran Associates, Inc. Hierarchical GUI benchmark from high-quality instructional videos.

[105] Lingfeng Bao, Jing Li, Z. Xing, Xinyu Wang, and Bo Zhou. Reverse engineering time-series interaction data from screen-captured videos. In *2015 IEEE 22nd International Conference on Software Analysis, Evolution, and Reengineering (SANER)*, pages 399–408, 2015.

[106] Alexandra List, Gala S Campos Oaxaca, Eunseo Lee, Hongcui Du, and Hye Yeon Lee. Examining perceptions, selections, and products in undergraduates' learning from multiple resources. In *The British journal of educational psychology*, 2021.

[107] Ching Liu, Juho Kim, and Hao-Chuan Wang. Conceptscape: Collaborative concept mapping for video learning. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, page 1–12, New York, NY, USA, 2018. Association for Computing Machinery.

[108] Guangyi Liu, Pengxiang Zhao, Liang Liu, Zhiming Chen, Yuxiang Chai, Shuai Ren, Hao Wang, Shibo He, and Wenchao Meng. Learnact: Few-shot mobile gui agent with a unified demonstration benchmark. *arXiv preprint arXiv:2504.13805*, 2025.

[109] Xingyu Liu, Patrick Carrington, Xiang 'Anthony' Chen, and Amy Pavel. What makes videos accessible to blind and visually impaired people? In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, New York, NY, USA, 2021. Association for Computing Machinery.

[110] Zipeng Liu, Zhicheng Liu, and Tamara Munzner. Data-driven multi-level segmentation of image editing logs. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, page 1–12, New York, NY, USA, 2020. Association for Computing Machinery.

[111] Yaxi Lu, Shenzhi Yang, Cheng Qian, Guirong Chen, Qinyu Luo, Yesai Wu, Huadong Wang, Xin Cong, Zhong Zhang, Yankai Lin, Weiwen Liu, Yasheng Wang, Zhiyuan Liu, Fangming Liu, and Maosong Sun. Proactive agent: Shifting LLM agents from reactive responses to active assistance. In *The Thirteenth International Conference on Learning Representations*, 2025.

[112] Justin Matejka, Tovi Grossman, and George Fitzmaurice. Ambient help. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, page 2751–2760, New York, NY, USA, 2011. Association for Computing Machinery.

[113] Justin Matejka, Tovi Grossman, and George Fitzmaurice. Patina: Dynamic heatmaps for visualizing application usage. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, page 3227–3236, New York, NY, USA, 2013. Association for Computing Machinery.

[114] Justin Matejka, Tovi Grossman, and George Fitzmaurice. Video lens: Rapid playback and exploration of large video collections and associated metadata. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST '14, page 541–550, New York, NY, USA, 2014. Association for Computing Machinery.

[115] Richard E. Mayer. *Cognitive Theory of Multimedia Learning*, page 31–48. Cambridge Handbooks in Psychology. Cambridge University Press, 2005.

[116] Matthew T. McCrudden, Ivar Bråten, and Ladislao Salmerón. *Learning from multiple texts*, pages 353–363. Elsevier, Netherlands, January 2022. Publisher Copyright: © 2023 Elsevier Ltd. All rights reserved.

[117] Paul F Merrill. Job and task analysis. In *Instructional technology: foundations*, 1987.

[118] Microsoft Corporation. Microsoft copilot. https://copilot.microsoft.com/, 2025. Accessed: 2025-11-14.

[119] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9876–9886, 2020.

[120] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *ICCV*, 2019.

[121] Davide Moltisanti, Frank Keller, Hakan Bilen, and Laura Sevilla-Lara. Learning Action Changes by Measuring Verb-Adverb Textual Relationships. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[122] Toni-Jan Keith Palma Monserrat, Shengdong Zhao, Kevin McGee, and Anshul Vikram Pandey. Notevideo: Facilitating navigation of blackboard-style lecture videos. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, page 1139–1148, New York, NY, USA, 2013. Association for Computing Machinery.

[123] Matt Morain and Jason Swarts. Yoututorial: A framework for assessing instructional online video. *Technical Communication Quarterly*, 21(1):6–24, 2012.

[124] Tushar Nagarajan and Lorenzo Torresani. Step differences in instructional video. In *CVPR*, 2024.

[125] Aadhavan M. Nambhi, Bhanu Prakash Reddy, Aarsh Prakash Agarwal, Gaurav Verma, Harvineet Singh, and Iftikhar Ahamath Burhanuddin. Stuck? no worries! task-aware command recommendation and proactive help for analysts. In *Proceedings of the 27th ACM Conference on User Modeling, Adaptation and Personalization*, UMAP '19, page 271–275, New York, NY, USA, 2019. Association for Computing Machinery.

[126] Mathieu Nancel and Andy Cockburn. Causality: A conceptual model of interaction history. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, page 1777–1786, New York, NY, USA, 2014. Association for Computing Machinery.

[127] Megha Nawhal, Jacqueline B. Lang, Greg Mori, and Parmit K. Chilana. Videowhiz: Non-linear interactive overviews for recipe videos. In *Proceedings of the 45th Graphics Interface Conference on Proceedings of Graphics Interface 2019*, GI'19, Waterloo, CAN, 2019. Canadian Human-Computer Communications Society.

[128] Shravan Nayak, Xiangru Jian, Kevin Qinghong Lin, Juan A. Rodriguez, Montek Kalsi, Nicolas Chapados, M. Tamer Özsu, Aishwarya Agrawal, David Vazquez, Christopher Pal, Perouz Taslakian, Spandana Gella, and Sai Rajeswar. Ui-vision: A desktop-centric gui benchmark for visual perception and interaction. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu, editors, *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 45817–45851. PMLR, 13–19 Jul 2025. Fine-grained desktop GUI benchmark with dense annotations.

[129] Cuong Nguyen and Feng Liu. Making software tutorial video responsive. CHI '15, page 1565–1568, New York, NY, USA, 2015. Association for Computing Machinery.

[130] Robert Nickerson, Upkar Varshney, and Jan Muntermann. A method for taxonomy development and its application in information systems. *European Journal of Information Systems*, 22, 05 2013.

[131] Donald A. Norman. *The Design of Everyday Things*. Basic Books, New York, 1988.

[132] Ioanna Ntinou, Enrique Sanchez, and Georgios Tzimiropoulos. Multiscale vision transformers meet bipartite matching for efficient single-stage action localization. In *CVPR*, 2024.

[133] Daulet Nurmanbetov. *BERT-restore-punctuation model from huggingface*, 2021.

[134] OpenAI. Function calling. https://platform.openai.com/docs/guides/function-calling, 2024. Accessed: Oct 9, 2024.

[135] OpenAI. Gpt-4o-2024-05-13. https://platform.openai.com/docs/models/gpt-4, 2024. Accessed: Oct 9, 2024.

[136] OpenAI. Gpt-4o system card, 2025.

[137] Hamid Palangi, Tomas Pfister, Yiwen Song, Luke Song, Oriana Riva, Palash Goyal, and Yu Su. Watch and learn: Learning to use computers from online videos. 2025.

[138] Junting Pan, Siyu Chen, Mike Zheng Shou, Jing Shao Yu Liu, and Hongsheng Li. Actor-context-actor relation network for spatio-temporal action localization. In *CVPR*, 2021.

[139] Amy Pavel, Dan B. Goldman, Björn Hartmann, and Maneesh Agrawala. Sceneskim: Searching and browsing movies using synchronized captions, scripts and plot summaries. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology*, UIST '15, page 181–190, New York, NY, USA, 2015. Association for Computing Machinery.

[140] Amy Pavel, Dan B. Goldman, Björn Hartmann, and Maneesh Agrawala. Vidcrit: Video-based asynchronous video review. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, UIST '16, page 517–528, New York, NY, USA, 2016. Association for Computing Machinery.

[141] Amy Pavel, Colorado Reed, Björn Hartmann, and Maneesh Agrawala. Video digests: A browsable, skimmable format for informational lecture videos. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST '14, page 573–582, New York, NY, USA, 2014. Association for Computing Machinery.

[142] SHRADDHA VIJAY PAWAR, Balavarun Pedapudi, Pramod Kaushik, Sarath Sivaprasad, Mario Fritz, and Shirish Karande. EARL: Early intent recognition in GUI tasks using theory of mind. In *ICML 2025 Workshop on Computer Use Agents*, 2025.

[143] Yi-Hao Peng, JiWoong Jang, Jeffrey P Bigham, and Amy Pavel. Say it all: Feedback for improving non-visual presentation accessibility. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, New York, NY, USA, 2021. Association for Computing Machinery.

[144] Yi-Hao Peng, Dingzeyu Li, Jeffrey P Bigham, and Amy Pavel. Morae: Proactively pausing ui agents for user choices. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology*, UIST '25, New York, NY, USA, 2025. Association for Computing Machinery.

[145] Raphaël Perraud, Aurélien Tabard, and Sylvain Malacria. Tutorial mismatches: investigating the frictions due to interface differences when following software video tutorials. In *Proceedings of the 2024 ACM Designing Interactive Systems Conference*, DIS '24, page 1942–1955, New York, NY, USA, 2024. Association for Computing Machinery.

[146] Chanda Phelan, Cliff Lampe, and Paul Resnick. It's creepy, but it doesn't bother me. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, page 5240–5251, New York, NY, USA, 2016. Association for Computing Machinery.

[147] Photoshop. *Adobe*, 2022 (accessed February 9, 2022).

[148] Suporn Pongnumkul, Mira Dontcheva, Wilmot Li, Jue Wang, Lubomir Bourdev, Shai Avidan, and Michael F. Cohen. Pause-and-play: Automatically linking screencast video tutorials with applications. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, page 135–144, New York, NY, USA, 2011. Association for Computing Machinery.

[149] Luca Ponzanelli, Gabriele Bavota, Andrea Mocci, Rocco Oliveto, Massimiliano Di Penta, Sonia Haiduc, Barbara Russo, and Michele Lanza. Automatic identification and classification of software

development video tutorial fragments. *IEEE Transactions on Software Engineering*, 45(5):464–488, 2019.

[150] Prolific. Prolific. https://www.prolific.co/, 2024. Accessed: Oct 9, 2024.

[151] Kevin Pu, Daniel Lazaro, Ian Arawjo, Haijun Xia, Ziang Xiao, Tovi Grossman, and Yan Chen. Assistance or disruption? exploring and evaluating the design and trade-offs of proactive ai programming support. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, CHI '25, New York, NY, USA, 2025. Association for Computing Machinery.

[152] Kevin Pu, Ting Zhang, Naveen Sendhilnathan, Sebastian Freitag, Raj Sodhi, and Tanya R. Jonker. Promemassist: Exploring timely proactive assistance through working memory modeling in multi-modal wearable devices. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology*, UIST '25, New York, NY, USA, 2025. Association for Computing Machinery.

[153] Alessandra Semeraro and Laia Turmo Vidal. Visualizing instructions for physical training: Exploring visual cues to support movement learning from instructional videos. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, New York, NY, USA, 2022. Association for Computing Machinery.

[154] Paul Hongsuck Seo, Arsha Nagrani, Anurag Arnab, and Cordelia Schmid. End-to-end generative pretraining for multimodal video captioning. In *CVPR*, 2022.

[155] Chirag Shah and Emily M. Bender. Situating search. In *Proceedings of the 2022 Conference on Human Information Interaction and Retrieval*, CHIIR '22, page 221–232, New York, NY, USA, 2022. Association for Computing Machinery.

[156] Yoav Shoham. *Reasoning about change: time and causation from the standpoint of artificial intelligence*. MIT Press, Cambridge, MA, USA, 1988.

[157] Tanmay Sinha, Patrick Jermann, Nan Li, and Pierre Dillenbourg. Your click decides your fate: Inferring information processing and attrition behavior from mooc video clickstream interactions. 09 2014.

[158] Yale Song, Gene Byrne, Tushar Nagarajan, Huiyu Wang, Miguel Martin, and Lorenzo Torresani. Ego4d goal-step: Toward hierarchical understanding of procedural activities. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023.

[159] Yixiao Song, Katherine Thai, Chau Minh Pham, Yapei Chang, Mazin Nadaf, and Mohit Iyyer. Bearcubs: A benchmark for computer-using web agents, 2025.

[160] Elizabeth S Spelke and Katherine D Kinzler. Core knowledge. *Developmental science*, 10(1):89–96, 2007.

[161] Yansong Tang, Dajun Ding, Yongming Rao, Yu Zheng, Danyang Zhang, Lili Zhao, Jiwen Lu, and Jie Zhou. Coin: A large-scale dataset for comprehensive instructional video analysis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[162] Qwen Team. Qwen3 technical report, 2025.

[163] Atima Tharatipyakul and Hyowon Lee. Towards a better video comparison: Comparison as a way of browsing the video contents. In *Proceedings of the 30th Australian Conference on Computer-Human Interaction*, OzCHI '18, page 349–353, New York, NY, USA, 2018. Association for Computing Machinery.

[164] TikTok. Tiktok, 2022 (accessed Sep 14, 2022).

[165] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In *NeurIPS*, 2022.

[166] Anh Truong, Floraine Berthouzoz, Wilmot Li, and Maneesh Agrawala. Quickcut: An interactive tool for editing narrated video. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, UIST '16, page 497–507, New York, NY, USA, 2016. Association for Computing Machinery.

[167] Anh Truong, Peggy Chi, David Salesin, Irfan Essa, and Maneesh Agrawala. Automatic generation of two-level hierarchical tutorials from instructional makeup videos. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, New York, NY, USA, 2021. Association for Computing Machinery.

[168] Teng Wang, Ruimao Zhang, Zhichao Lu, Feng Zheng, Ran Cheng, and Ping Luo. End-to-end dense video captioning with parallel decoding. In *ICCV*, 2021.

[169] Weiying Wang, Yongcheng Wang, Shizhe Chen, and Qin Jin. Youmakeup: A large-scale domain-specific multimodal dataset for fine-grained semantic comprehension. In *EMNLP-IJCNLP*, 2019.

[170] Xu Wang, Benjamin Lafreniere, and Tovi Grossman. Leveraging community-generated videos and command logs to classify and recommend software workflows. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, page 1–13, New York, NY, USA, 2018. Association for Computing Machinery.

[171] Yi Wang, Yinan He, Yizhuo Li, Kunchang Li, Jiashuo Yu, Xin Ma, Xinyuan Chen, Yaohui Wang, Ping Luo, Ziwei Liu, Yali Wang, Limin Wang, and Yu Qiao. Internvid: A large-scale video-text dataset for multimodal understanding and generation. *ICLR*, 2024.

[172] Yi Wang, Xinhao Li, Ziang Yan, Yinan He, Jiashuo Yu, Xiangyu Zeng, Chenting Wang, Changlian Ma, Haian Huang, Jianfei Gao, Min Dou, Kai Chen, Wenhai Wang, Yu Qiao, Yali Wang, and Limin Wang. Internvideo2.5: Empowering video mllms with long and rich context modeling. *arXiv preprint arXiv:2501.12386*, 2025.

[173] Min web browser. *Min browser*, 2022 (accessed February 9, 2022).

[174] Whale web browser. *Naver*, 2022 (accessed February 9, 2022).

[175] Sarah Weir, Juho Kim, Krzysztof Z. Gajos, and Robert C. Miller. Learnersourcing subgoal labels for how-to videos. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, CSCW '15, page 405–416, New York, NY, USA, 2015. Association for Computing Machinery.

[176] WikiHow. Wikihow. https://www.wikihow.com/, 2024. Accessed: Oct 9, 2024.

[177] Shirley Wu, Michel Galley, Baolin Peng, Hao Cheng, Gavin Li, Yao Dou, Weixin Cai, James Zou, Jure Leskovec, and Jianfeng Gao. Collabllm: From passive responders to active collaborators. In *International Conference on Machine Learning (ICML)*, 2025.

[178] Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. Next-qa: Next phase of question-answering to explaining temporal actions. In *CVPR*, 2021.

[179] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *CVPR*, 2016.

[180] Kuldeep Yadav, Ankit Gandhi, Arijit Biswas, Kundan Shrivastava, Saurabh Srivastava, and Om Deshmukh. Vizig: Anchor points based non-linear navigation and summarization in educational videos. In *Proceedings of the 21st International Conference on Intelligent User Interfaces*, IUI '16, page 407–418, New York, NY, USA, 2016. Association for Computing Machinery.

[181] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Just ask: Learning to answer questions from millions of narrated videos. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1666–1677, 2020.

[182] Antoine Yang, Arsha Nagrani, Paul Hongsuck Seo, Antoine Miech, Jordi Pont-Tuset, Ivan Laptev, Josef Sivic, and Cordelia Schmid. Vid2seq: Large-scale pretraining of a visual language model for dense video captioning. In *CVPR*, 2023.

[183] Bufang Yang, Lilin Xu, Liekang Zeng, Kaiwei Liu, Siyang Jiang, Wenrui Lu, Hongkai Chen, Xiaofan Jiang, Guoliang Xing, and Zhenyu Yan. Contextagent: Context-aware proactive llm agents with open-world sensory perceptions, 2025.

[184] Qinglong Yang, Haoming Li, Haotian Zhao, Xiaokai Yan, Jingtao Ding, Fengli Xu, and Yong Li. Fingertip 20k: A benchmark for proactive and personalized mobile llm agents, 2025.

[185] Saelyne Yang. Enhancing how people learn procedural tasks through how-to videos. In *Adjunct Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, UIST Adjunct '24, New York, NY, USA, 2024. Association for Computing Machinery.

[186] Saelyne Yang and Juho Kim. What makes it hard for users to follow software tutorial videos? In *Proceedings of HCI Korea 2020*, pages 531–536, South Korea, 2020. The HCI Society of KOREA.

[187] Saelyne Yang, Sangkyung Kwak, Tae Soo Kim, and Juho Kim. Improving video interfaces by presenting informational units of videos. In *CHI'22 Extended Abstracts*. Association for Computing Machinery, 2022.

[188] Saelyne Yang, Sangkyung Kwak, Juhoon Lee, and Juho Kim. Beyond instructions: A taxonomy of information types in how-to videos. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, New York, NY, USA, 2023. Association for Computing Machinery.

[189] Saelyne Yang, Sunghyun Park, Yunseok Jang, and Moontae Lee. YTCommentQA: Video Question Answerability in Instructional Videos. In *AAAI*, 2024.

[190] Saelyne Yang, Anh Truong, Juho Kim, and Dingzeyu Li. Videomix: Aggregating how-to videos for task-oriented learning. In *Proceedings of the 30th International Conference on Intelligent User Interfaces*, IUI '25, page 1564–1580, New York, NY, USA, 2025. Association for Computing Machinery.

[191] Saelyne Yang, Jisu Yim, Aitolkyn Baigutanova, Seoyoung Kim, Minsuk Chang, and Juho Kim. Softvideo: Improving the learning experience of software tutorial videos with collective interaction data. In *27th International Conference on Intelligent User Interfaces*, IUI '22, page 646–660, New York, NY, USA, 2022. Association for Computing Machinery.

[192] Saelyne Yang, Jisu Yim, Juho Kim, and Hijung Valentina Shin. Catchlive: Real-time summarization of live streams with stream content and interaction data. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, New York, NY, USA, 2022. Association for Computing Machinery.

[193] Suyu Ye, Haojun Shi, Darren Shih, Hyokun Yun, Tanya Roosta, and Tianmin Shu. Realwebassist: A benchmark for long-horizon web assistance with real-world users. *arXiv preprint arXiv:2504.10445*, 2025.

[194] YouTube. Video chapters, 2022 (accessed Sep 14, 2022).

[195] youtube dl. *youtube-dl*, 2022 (accessed Sep 14, 2022).

[196] Abhay Zala, Jaemin Cho, Satwik Kottur, Xilun Chen, Barlas Oğuz, Yashar Mehdad, and Mohit Bansal. Hierarchical video-moment retrieval and step-captioning. In *CVPR*, 2023.

[197] Bofei Zhang, Zirui Shang, Zhi Gao, Wang Zhang, Rui Xie, Xiaojian Ma, Tao Yuan, Xinxiao Wu, Song-Chun Zhu, and Qing Li. Tongui: Building generalized gui agents by learning from multimodal web tutorials. *arXiv preprint arXiv:2504.12679*, 2025. Introduces the TongUI framework and GUI-Net-1M dataset.

[198] Ceyao Zhang, Kaijie Yang, Siyi Hu, Zihao Wang, Guanghe Li, Yihang Sun, Cheng Zhang, Zhaowei Zhang, Anji Liu, Song-Chun Zhu, Xiaojun Chang, Junge Zhang, Feng Yin, Yitao Liang, and Yaodong Yang. Proagent: Building proactive cooperative agents with large language models, 2024.

[199] Jiwen Zhang, Jihao Wu, Teng Yihua, Minghui Liao, Nuo Xu, Xiao Xiao, Zhongyu Wei, and Duyu Tang. Android in the zoo: Chain-of-action-thought for GUI agents. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 12016–12031, Miami, Florida, USA, November 2024. Association for Computational Linguistics.

[200] Lei Zhang, Qian-Kun Xu, Lei-Zheng Nie, and Hua Huang. Videograph: a non-linear video representation for efficient exploration. *Vis. Comput.*, 30(10):1123–1132, October 2014.

[201] Henry Hengyuan Zhao, Kaiming Yang, Wendi Yu, Difei Gao, and Mike Zheng Shou. Worldgui: An interactive benchmark for desktop gui automation from any starting point, 2025.

[202] Wentian Zhao, Seokhwan Kim, Ning Xu, and Hailin Jin. Video question answering on screencast tutorials. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, IJCAI'20, 2021.

[203] Yaxi Zhao, Razan Jaber, Donald McMillan, and Cosmin Munteanu. "rewind to the jiggling meat part": Understanding voice control of instructional videos in everyday tasks. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, New York, NY, USA, 2022. Association for Computing Machinery.

[204] Yuheng Zhao, Xueli Shu, Liwen Fan, Lin Gao, Yu Zhang, and Siming Chen. Proactiveva: Proactive visual analytics with llm-based ui agent, 2025.

[205] Luowei Zhou, Chenliang Xu, and Jason J Corso. Towards automatic learning of procedures from web instructional videos. In *AAAI Conference on Artificial Intelligence*, pages 7590–7598, 2018.

[206] Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Yonatan Bisk, Daniel Fried, Uri Alon, et al. Webarena: A realistic web environment for building autonomous agents. *arXiv preprint arXiv:2307.13854*, 2023.

[207] Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

[208] Dimitri Zhukov, Jean-Baptiste Alayrac, Ramazan Gokberk Cinbis, David Fouhey, Ivan Laptev, and Josef Sivic. Cross-task weakly supervised learning from instructional videos. In *CVPR*, 2019.

Acknowledgment

My PhD journey and this thesis would not have been possible without the unwavering support of many people. Throughout my doctoral studies, I was surrounded by encouragement, guidance, and care that carried me through both rewarding and challenging moments.

First and foremost, I would like to express my deepest gratitude to **my advisor, Juho Kim**. I still vividly remember our conversations about the vision of videos when I began my PhD. Since then, I have learned immensely from you. Your critical yet forward-looking advice has shaped my research, and your guidance has strongly influenced how I approach research with a long-term perspective. Thank you for your patience during moments when I lost direction, and for your constant support that helped shape me as a researcher today.

I am also deeply thankful to **my thesis committee members**, **Tak Yeon Lee, Joon Son Chung, Amy Pavel, and Yale Song**. Your thoughtful feedback significantly strengthened this thesis and pushed me to consider broader implications and impact beyond individual projects. I truly appreciate your support and encouragement throughout this process.

I would like to thank **all past and current members of KIXLAB** for creating such a warm, collaborative, and inspiring environment. **Yoonseo**, thank you for your constant care and support that helped me through many ups and downs. **Tae Soo**, your insightful feedback greatly helped my research. **Yoonsu**, your energy and enthusiasm made life in the lab enjoyable. **DaEun**, your humor and charismatic presence made the lab feel welcoming and fun. **Kihoon**, thank you for pulling me out for coffee and dinner. **Yoonjoo**, my internship and conference mate, I truly enjoyed the time we spent together. **Hyunwoo**, thank you for handling so many behind-the-scenes tasks in the lab. You will always be the lab master to me. **Bekzat**, my long-term collaborator, I deeply appreciated your thoughtful discussions and how we turned ideas into meaningful research. **Jaesang**, I really enjoyed exploring diverse ideas together, and it has been a joy to see your growth as a researcher. **Eunyoung, Seoyoung, Hyungyu, and Minsuk**, I am grateful for the time we shared in the lab and for all the advice and conversations along the way. I am forever grateful to have been part of such a welcoming and supportive community.

I would also like to thank my collaborators, including **Jisu, Sangkyung, Juhoon, Aitolkyn, Kevin, and Jae Won**, for shaping my research through valuable discussions and constructive feedback. I am also grateful to my **internship mentors, Valentina Shin, Ding Li, Anh Truong, Jo Vermeulen, Justin Matejka, George Fitzmaurice, Sunghyun Park, Yunseok Jang, and Moontae Lee**, for their guidance and mentorship, which broadened my perspective on research. I would like to thank the **friends and colleagues** I met through internships, conferences, and both inside and outside the school. You made the research journey joyful and memorable. In particular, I would like to thank our Seattle Squad, **Mina and Yi-Hao**. I am deeply grateful for the friendship we built and will always cherish the time we shared.

I am deeply grateful to **the KCIC community**, which supported and sustained me throughout my graduate studies. I am sincerely grateful to **Prof. Cho, Mrs. Park, Prof. Nam, and Prof. Ka**, for their unwavering support and prayers. During times of struggle, you reminded me of what truly matters in life. I also thank **Yeonju, Sunhyoung, Natasha, Pei Jia, and other friends at KCIC** who stood by me and encouraged me. You made this journey more humane and meaningful. I am also thankful to **Prof. Uichin Lee** for his support and prayers throughout the time at KAIST.

Lastly, I express my deepest gratitude to my family. I thank **my parents, Jaejun and Eunhui**, for always believing in me and for their endless love. It is because of you that I am here today. To **my sister, Yaelyne**, thank you for always standing by my side. I am also grateful to **my parents-in-law, Mu Yeol and Hyun Young**, for constant encouragement and support. To **my husband, Junyong**, your unconditional love was the foundation that carried me through every stage of this journey. You are my best teammate in both research and life, complementing and supporting me in every way. From our undergraduate days to defending our dissertations on the same day, thank you for sharing every moment of this journey with me. Above all, I thank God for His guidance and love. I hope to pass on the love I have received to the world around me.

Curriculum Vitae

Education

2021. 3. – 2026. 2.	Ph.D., School of Computing, KAIST
2019. 3. – 2021. 2.	M.S., School of Computing, KAIST
2015. 3. – 2019. 2.	B.S., School of Computing, KAIST

Employment

2024. 5. – 2024. 8.	Research Intern, Adobe Research
2023. 6. – 2023. 9.	Research Intern, Autodesk Research
2022. 10. – 2023. 2.	Research Intern, LG AI Research
2020. 6. – 2020. 9.	Research Intern, Adobe Research

Publications

1. **Saelyne Yang**, Anh Truong, Juho Kim, Dingzeyu Li. "VideoMix: Aggregating How-To Videos for Task-Oriented Learning", *IUI 2025: ACM Conference on Intelligent User Interfaces*
2. **Saelyne Yang**, Jo Vermeulen, George Fitzmaurice, Justin Matejka. "AQuA: Automated Question-Answering in Software Tutorial Videos with Visual Anchors", *CHI 2024: ACM Conference on Human Factors in Computing Systems*
3. **Saelyne Yang**, Sunghyun Park, Yunseok Jang, Moontae Lee. "YTCommentQA: Video Question Answerability in Instructional Videos", *AAAI 2024: Association for the Advancement of Artificial Intelligence*
4. **Saelyne Yang**. "Enhancing How People Learn Procedural Tasks Through How-to Videos", *UIST 2024 Doctoral Symposium*
5. **Saelyne Yang**, Jaesang Yu, Jae Won Cho, Juho Kim. "Fine-Grained Action Understanding with Tools in Instructional Videos", *CVPR 2024 Workshop on Learning from Procedural Videos and Language*
6. **Saelyne Yang**, Sangkyung Kwak*, Juhoon Lee*, Juho Kim. "Beyond Instructions: A Taxonomy of Information Types in How-to Videos", *CHI 2023: ACM Conference on Human Factors in Computing Systems*
7. **Saelyne Yang**, Jisu Yim, Aitolkyn Baigutanova, Seoyoung Kim, Minsuk Chang, Juho Kim. "SoftVideo: Improving the Learning Experience of Software Tutorial Videos with Collective Interaction Data", *IUI 2022: ACM Conference on Intelligent User Interfaces*
8. **Saelyne Yang**, Jisu Yim, Juho Kim, Hijung Valentina Shin. "CatchLive: Real-time Summarization of Live Streams with Stream Content and Interaction Data", *CHI 2022: ACM Conference on Human Factors in Computing Systems*

9. **Saelyne Yang**, Sangkyung Kwak, Tae Soo Kim, Juho Kim. "Improving Video Interfaces by Presenting Informational Units of Videos", *CHI 2022 Workshop on Computational Approaches for Understanding, Generating, and Adapting User Interfaces*

10. **Saelyne Yang**, Changyoon Lee, Hijung Valentina Shin, Juho Kim. "Snapstream: Snapshot-based Interaction in Live Streaming for Visual Art", *CHI 2020: ACM Conference on Human Factors in Computing Systems*

11. **Saelyne Yang**, Juho Kim. "What Makes It Hard for Users to Follow Software Tutorial Videos?", *HCI Korea 2020, The HCI Society of Korea*