A Text-Native Interface for Generative Video Authoring

Xingyu Bruce Liu
Adobe Research
San Francisco, CA, USA
xingyul@adobe.com

Mira Dontcheva
Adobe Research
Seattle, WA, USA
mirad@adobe.com

Dingzeyu Li
Adobe Research
Seattle, WA, USA
dinli@adobe.com

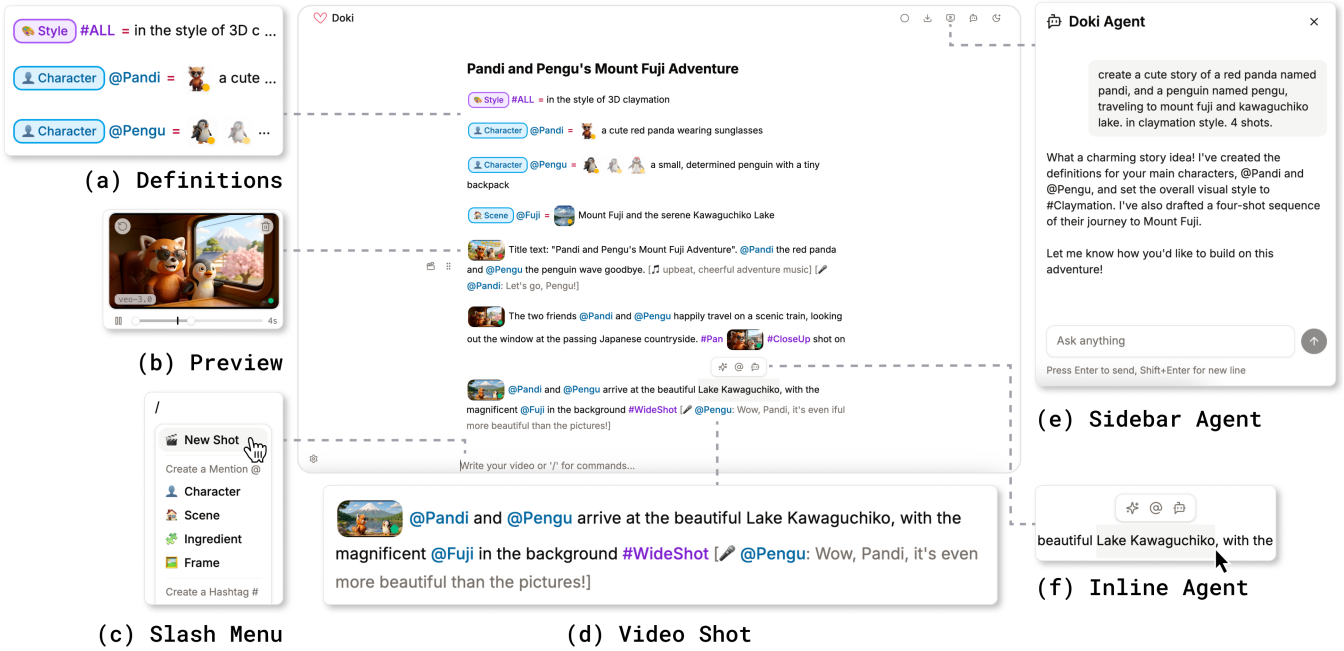

Figure 1: *Doki* is a text-native video authoring interface that creates generative videos within a single document. (a) Users define reusable assets and styles with mentions and hashtags, (b) view and adjust inline previews, (c) access commands through a slash menu, (d) write paragraphs that compile into video shots, and collaborate with both (e) a conversational AI agent and (f) inline agent.

Abstract

Everyone can write their stories in freeform text format – it's something we all learn in school, yet authoring video requires one to learn specialized and complicated tools. In this paper, we introduce *Doki*, a text-native interface for generative video authoring. In Doki, writing text is the primary interaction: within a single document, users define assets, structure scenes, generate shots, refine edits, and add audio. We articulate the design principles of this text-native approach and demonstrate Doki's capabilities through examples. We ran a week-long diary study with 10 participants of varied expertise. Participants produced 46 videos, reporting faster idea-to-draft flow, improved coherence through parameterization, and clearer comprehension of narrative structure in the document view; they also surfaced limitations around model predictability, precise control, and temporal expressivity. With Doki, we explore a fundamental shift in generative video interfaces, and demonstrate a powerful and accessible new way to craft visual stories.

CCS Concepts

• **Human-centered computing** → **Interactive systems and tools**; **Natural language interfaces**.

Keywords

Generative AI, Generative Video, AI-Generated Video, Video, Text-to-Video, Video Editing, Creativity Support

ACM Reference Format:
Xingyu Bruce Liu, Mira Dontcheva, and Dingzeyu Li. 2025. A Text-Native Interface for Generative Video Authoring. In . ACM, New York, NY, USA, 22 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

Conference'17, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Writing gives form to our imagination and stories. Children are taught the basics of narrative structure - beginning, middle and end – as early as first grade. Over time, writing text turns into a

core means for communication and self-expression, as we compose emails, articles, or text messages. With the rise of video platforms like YouTube and TikTok, video has emerged as a new powerful alternative for communication, expression, and storytelling. And recent advances in generative AI introduce new possibilities for content creation. As a result, more and more people are authoring video.

Unfortunately, authoring video remains challenging, because it relies on a very different set of tools than the ones we learn in school. Traditional video non-linear editors emphasize rich interfaces with many capabilities excelling at fine-grained editing tasks at the cost of usability and accessibility. Modern editors, such as Descript [12] reduce friction by supporting text-based editing with transcript text, yet such tools only use text for editing speech tracks and adopt the experience of traditional tools with multiple panes and multi-track timelines for authoring visuals.

Generative AI introduces a paradigm shift in creative tools. Many call generative models "the new camera", because they let people produce high-quality content with a simple text prompt. In this new era, **text is again a central medium for creative expression**. While most generative video tools [6, 15, 21, 27, 29] focus solely on individual clips and still rely on traditional workflows and timelines for narrative storytelling, we ask: if audiovisual content can be generated directly from text, **can video authoring become as natural as editing a document?**

We present *Doki*, a text-native interface for authoring generative videos[1]. In Doki, writing is the primary interaction. Within a **single document**, users define assets, structure scenes, generate shots, refine edits, and add audio. The document serves both as narrative and as an executable script for video production. Our design pursues four goals:

(1) *Make text a central medium for authoring.* Text is both natural for people and native to AI. It serves as a perfect intermediate common ground between human and AI, as it allows humans to freely edit and quickly review, while enabling AI to generate and suggest using the same medium. Its freeform nature also supports flexible workflows that move fluidly between ideation, generation, and revision.

(2) *Consolidate authoring into a single representation.* Current workflows require creators to juggle multiple forms and views of the same video, increasing cognitive load [8]. In Doki, scripts, prompts, visuals, audio, and timelines co-exist in a structured text representation. While we do not aim for a full replacement for professional toolchains, Doki supports many end-to-end creative tasks.

(3) *Preserve consistency via parameterization.* As projects scale, maintaining coherence becomes difficult. Doki introduces parametrized definitions to keep characters, styles, and other elements consistent across narratives, and makes it easier to construct complex scenes and develop longer, more cohesive stories.

(4) *Keep interaction simple.* Professional tools are powerful but can be overwhelming with complex UI. Doki offers a minimalist design with a lightweight slash menu for creating assets and shots, and inline shot previews for reviewing and editing directly in the document.

We evaluated Doki in a week-long diary study with 10 participants, spanning filmmakers to first-time creators. Participants submitted 46 videos and on average rated Doki System Usability Scale 81.2 ("Excellent"). They reported that Doki's text-native structure not only accelerated the transition from ideation to output, but also enhanced their holistic understanding of narrative flow. Furthermore, participants found that parameterized definitions provides a reliable structural backbone for cohesive storytelling. Our findings also reveal that the system's utility scaled distinctly with user expertise. Novices felt newly empowered to author visual stories that were previously out of reach. Conversely, domain experts integrated the tool as a complementary engine for rapid ideation and storyboarding, rather than a replacement for high-fidelity production environments.

The study also surfaced insights into human-AI collaboration. The document interface encouraged users to heavily delegate production tasks to AI agents. Yet, despite this extensive automation, creators maintained a strong sense of authorship, frequently likening their role to a "director". Text document serves as a perfect "common ground", an intermediate representation that is simultaneously readable and editable by both parties. Humans can freely build the narrative foundation, while AI operations, whether generating drafts or applying edits, remain transparent, legible, and fully revisable by the human.

In summary, we contribute:

- **A structured, parametrized text-based representation for videos**. We propose a three-layer, hierarchical *"document as video, paragraph as sequence, sentence as shot"* structure, and introduces a definition system (@mentions and #hashtags) with propagation and context handling to ensure cross-shot consistency. This representation is independent of any particular interface.
- **Doki, a minimalistic, text-native interface** that instantiates this representation. Users can create videos end-to-end, from ideation to export, within one single document. Doki employs a minimalistic user interface design with a lightweight slash menu, inline previews and AI agents.
- **An in-the-wild evaluation of Doki**. Empirical insights into human-AI collaboration dynamics, demonstrating how a text-native interface shifts creator workflows, balances the trade-off between macro-level storytelling and micro-temporal control, and negotiates the boundaries of artistic agency and automation .

2 Related Work

Our work is situated at the intersection of three research areas: video generation models, novel interfaces for video creation, and HCI research on dynamic documents.

2.1 Generative Models as Individual Video Generators

Recent advances in generative AI have produced models capable of synthesizing high-fidelity video clips from text descriptions. Commercially available systems like Runway [29], Pika [27], Kling [21]

[1]We use the term *generative videos* to denote content produced with generative AI methods. Unlike "AI-generated videos," which can imply AI-led authorship, *generative videos* is neutral about agency and ownership.

(a) "Bento Box" like interfaces (b) *Doki*'s approach

Figure 2: A comparison of interface paradigms. (a) "Bento Box" style interfaces distribute authoring across multiple, separate representations. (b) Doki's approach uses a text-native canonical representation where the document serves as the primary interface.

and Google's Veo [15] demonstrate increasingly powerful capabilities. The primary interaction model for these tools is to write a text prompt to generate a single, short, fixed-length video. These prompts often require a structured format, specifying elements like the subject, action, context, and desired visual style [15]. More advanced features allow for greater control, such as using reference images to guide generation or maintain character consistency [29], and using start and end frames to define transitions [15, 24, 27].

However, the focus of these models remains on the generation of *individual, short clips*, typically lasting only a few seconds, *e.g.*, 8 seconds for Veo 3. There is limited to no support for structuring these shots into a longer, coherent narrative. Users must manually assemble the generated clips in a separate editing environment. Doki builds upon the capabilities of such models. It treats them as the supporting engine while contributing a higher-level authoring interface that shifts the interaction from generating isolated clips to authoring a complete video story.

2.2 Novel Video Authoring Interfaces

A growing body of HCI research explores AI-powered interfaces for video authoring beyond traditional non-linear editors. Videorigami, for example, organizes authoring around multiple *compositional structures* (*e.g.,* canvas, script, storyboard, timeline) and synchronizes them to support human–AI co-creation [7]. LAVE augments editing with AI agent and language annotations of footage [35]. ExpressEdit lets creators issue multimodal edit intents by combining natural language and on-frame sketching [32]. While these systems explore new interaction modalities, they typically distribute authoring across multiple panes – what we refer to as a "bento box" interface (Figure 2a) – which can incur split-attention costs when creators must reconcile multiple views and formats [9].

Doki embraces a contrasting design philosophy aimed at consolidating all authoring steps into a single representation.

Subtractive vs. additive workflows. Due to the nature of video capture prior to generative video models, most existing tools adopt *subtractive* workflows: creators begin with pre-recorded footage (or a library) and remove, trim, and re-order material. Examples include QuickCut [33]; transcript-aligned manipulation of talking-head footage [14]; and multimodal edit commands that operate on existing clips [32]. Some systems explore *additive* workflows where the tool synthesizes assets from text, such as Doc2Video's document-to-talking-head prototyping [10], but their system is limited to specific shot types rather than authoring complete, multi-shot stories. Doki takes an additive stance end-to-end: text is the substrate for *generating* shots and assets, not only selecting or trimming them.

Transcript-based editing. Transcript-centered tools let users edit a video by editing its text transcript, with changes reflected on the timeline [10, 12, 17, 28]. This interaction is especially effective for dialogue-driven media such as interviews and podcasts, where transcript and timeline are tightly aligned. Other systems use text to guide the assembly of existing footage: Write-A-Video [36] selects shots that semantically match an edited paragraph, while Crosscast and Crosspower augment audio with retrieved images [37, 38]. Doki complements these approaches by extending the role of text from mirroring transcript to *generating the full video* including audio tracks and video frames. Rather than treating text as just a proxy to transcript/speech tracks, Doki uses text to create and manage the visuals—and associated audio—end-to-end.

2.3 Rich-Text Editing and Dynamic Documents

Doki's interface design is inspired by research on dynamic documents [2, 18, 19, 23, 34]. Unlike applications that enforce predefined feature sets and rigid task boundaries, dynamic documents are designed as permissive mediums. Much like writing on paper, they do not impose a strict schema, allowing users to capture ideas in freeform ways. This flexibility supports a workflow of "gradual enrichment" [19], where users can begin with unstructured text and progressively add structure and computational behavior as their needs evolve. Research prototypes such as Potluck [19] and Embark [18] demonstrate this principle by allowing notes to evolve into personal software or interactive trip plans.

Commercial tools have adopted related approaches. Notion [25] provides a block-based workspace where users can draft in natural text and later enrich content. OpenAI's Canvas mode [26] illustrates a similar trajectory. It extends chat-based interaction into an editable document environment where users can directly edit and request inline assistance.

Doki builds on these precedents but extends them into the video authoring domain. It treats video creation as the authoring of dynamic documents, that can be read as narratives and executed as a script, supporting a more flexible authoring process than existing video tools.

3 How People Create Generative Videos Today

We examined current end-to-end workflows for creating generative videos and describe motivating scenarios for Doki. Our analysis drew on public creators' generative AI video tutorials on YouTube and X [1, 20, 30, 31], and on observations of creators within our organization.

3.1 Motivating Scenarios

3.1.1 Creator A: Asset-First. Creator A anchored storytelling in asset construction. She typically spent 3-5 hours creating characters with text-to-image models and used the results as reference frames for consistency across a project. Yet, consistency remained elusive

even with reference images. As she explained, *"I actually had two different octopus character reference frames that I used in GPT Image when I wanted more or less exhausted."* Because outputs often diverged, she repeatedly checked generation outputs and manually adjusted assets. After preparing characters and environments, she created storyboards, generated short clips, and finally imported everything into a separate timeline editor to add voice and music. Her workflow required constant switching between asset creation tools, storyboard generators, video models, and editing software. Each step relied on a different view or interface, forcing repeated translation between formats.

3.1.2 Creator B: Script-First. Creator B started with a complete script – often drafted with a large language model – and decomposed it into scenes. Each scene was expanded into a detailed prompt specifying setting, character, tone, and cinematographic parameters. Because current video models lacked context awareness, every shot required re-describing all elements to preserve consistency. He explained, *"I would re-describe the setting, the character, and the tone every time."* This led to long and repetitive prompts, rendering small edits nearly impossible. Once clips were generated, they were reviewed and assembled in a video editor. This process prioritized prompt construction over narrative development.

3.1.3 Creator C: Iterative-Exploratory. Creator C worked in a highly exploratory style. Rather than preparing a script or asset library, he generated clips rapidly and used them to guide the next step. This improvisational process enabled serendipitous discoveries but became fragile during revision. For example, shifting the story setting from Tokyo to Barcelona required revisiting every prompt and regenerating all clips. As with others, he then manually reassembles on a timeline, further slowing iteration.

3.1.4 Recurring Challenges in Generative Video Authoring. Although creators approached generative video authoring very differently, many encountered the same set of challenges:

C1. Fragmented Tools and Formats: Even for a simple short video, many creators end up using three to five different tools — for example, a text editor for scripts, an image generator for visuals, a video model for clips, audio tools for narration or music, and a non-linear editor to pull it all together. The core elements of a project — scripts, prompts, images, audio, and edits — often live in separate formats across these tools.

This fragmentation can lead to constant context-switching, breaking creative flow and making it harder to maintain momentum. It also makes it difficult to get a clear overview of the project or understand how one change affects others. This raises the barrier to entry and slows iteration.

C2. Prompt Over Storytelling: Creators often spend disproportionate time crafting and tweaking prompts, which shifts focus away from narrative development and toward prompt engineering. For example, a 30-shot video might require 30 separate, verbose prompts – often managed manually in a list. Building up a complex prompt is itself difficult, and even once crafted, it is hard to control or manage across shots.

As a result, the workflow becomes less about shaping a story and more about wrestling with prompts, leaving creators with less bandwidth for the higher-level creative choices that should drive their work.

C3. Consistency and Coherence: Maintaining visual and narrative consistency remains difficult. While recent models such as Nano Banana [13] and Flux Kontext [4] can generate consistent images from references, we argue that true coherence is as much an authoring and representation problem as it is a modeling problem. Conditioning on a few reference images does not provide the structural backbone needed to preserve assets, styles, and relationships across an extended narrative. Without a representation that defines elements, propagates them through time, and manages context automatically, consistency will inevitably drift as projects grow.

3.2 Design Principles

Our analysis of existing workflows reveals a clear opportunity for a new authoring paradigm for generative videos. We propose four core design principles that guide the creation of Doki:

D1. Make text the central medium for authoring. C2 shows how prompt engineering can pull attention away from storytelling. Doki addresses this by making text the primary medium for authoring: prompts are embedded directly into the narrative, so the creative process feels more like writing a story than managing a command list.

Text has several advantages as a medium: it is familiar and accessible for humans, the native input modality for generative models, and inherently flexible and free-form. Because both humans and AI operate in the same medium, revisions and suggestions are immediately understandable: for example, AI-generated revisions are immediately understandable and can be refined in-place by the user, enabling fast and straightforward iteration.

D2. Consolidate video authoring in a single representation. In traditional workflows, creators switch between multiple views of the same project, which fragments the process and increases cognitive load (C1). Doki instead unifies scripts, prompts, visuals, audio, and timelines into a single canonical text representation. This representation functions both as the editable document that authors work with and as the underlying schema for the video itself. While not a full replacement for professional toolchains, this unified form enables many end-to-end creative tasks without leaving the document.

D3. Preserve consistency through parameterization. As projects scale, maintaining coherence across characters, styles, and scenes becomes increasingly difficult (C2, C3). Doki addresses this with a system of parameterized definitions that ensures elements remain consistent while still allowing flexible variation. In addition, Doki automatically retrieves relevant references from earlier scenes so that continuity is preserved across the narrative. Together, these mechanisms make it easier to construct complex scenes, reuse components, and extend projects into longer, more cohesive stories.

D4. Keep interaction simple. Traditional non-linear editors are powerful but can be overwhelming (C1). Doki offers a simple editing environment with a lightweight slash menu for generating shots, defining assets, and inserting audio. Authors can write, generate, and edit within the paragraph, while using inline previews to review and trim.

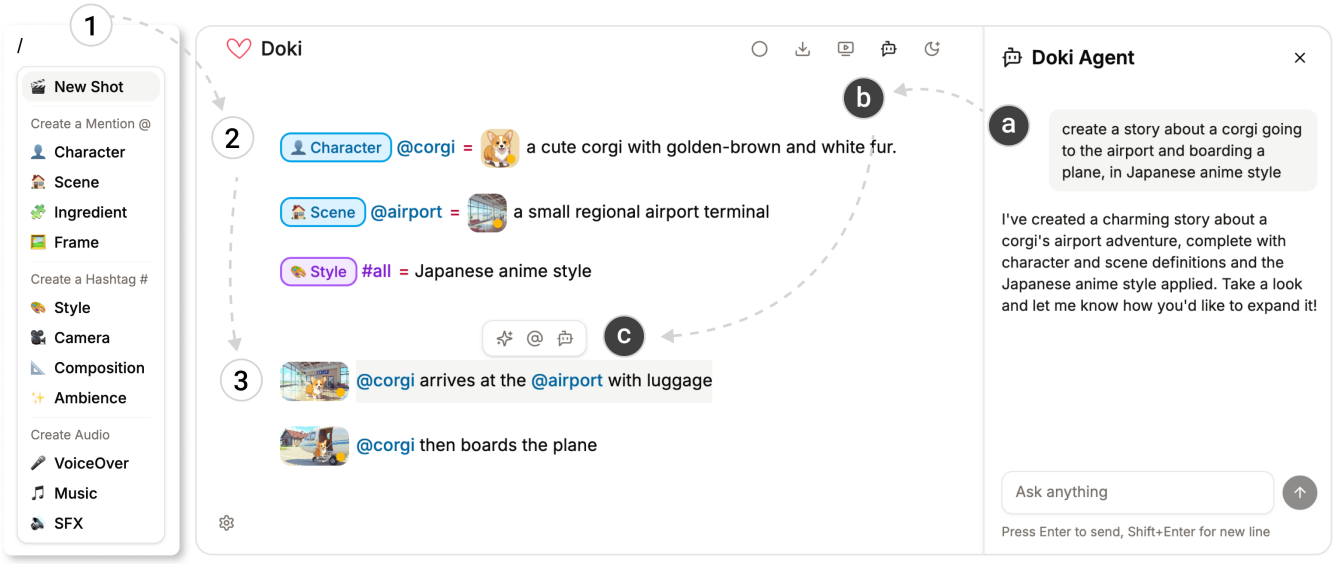

Figure 3: Two basic example workflows in Doki. Alice: (1) define assets and shots with slash commands → (2) write story and generate previews → (3) create video shots; Bob: (a) prompt the sidebar agent for a draft → (b) review the AI-generated draft → (c) refine with inline AI agent.

4 Doki

In this section, we first show example workflows of using Doki, then describe the structured text representation. Finally, we walk through each component of the system.

4.1 Example Workflows

Two creators produce the same short animation in different ways (Figure 3). The story follows a cute little corgi embarking on an adventure.

4.1.1 Alice: Writing a Video from Scratch. Alice prefers a hands-on, deliberate process. Starting with a blank text editor, she first defines the main assets for her story using slash commands. She types `/` and selects 👤 Character from the menu to create a definition block. Here, she names the character @corgi and provides a textual description: "a cute corgi with golden-brown and white fur". Similarly, she defines the setting with `/` → 🏠 Scene, naming it @airport and describing it as "a small regional airport terminal". To ensure a consistent visual aesthetic, she creates a global style definition using `/` → 🎨 Style, naming it #all and adding her description. For each definition, Alice generates a preview image. This allows her to check the visual interpretation and use the image as a reference to maintain consistency.

Next, Alice writes the storyline directly in the document. She structures it in paragraphs each corresponding to a video clip. For each clip, she creates a shot by selecting new shot from the slash menu (`/` → 🎬 New Shot). She writes the first shot: @corgi arrives at the @airport with luggage, and the second shot: @corgi then boards the plane. She then generates an image preview for each shot, following by the video. Once all shots are ready, she can export the video or view it in the built-in player.

4.1.2 Bob: Directing AI agents. In contrast, Bob prefers to leverage an AI agent workflow. Instead of starting from scratch, Bob uses the sidebar AI agent to generate the initial script. He provides a one-liner prompt: "a story about a corgi going to the airport and boarding a plane, in Japanese anime style". The agent generates a complete document structure, including definitions for the character, scene, and style, along with two paragraphs of text for the initial shots.

To refine the agent's output, Bob can manually edit the document and the description of @corgi. Or, he can invoke the inline AI agent. For example, he selects the first paragraph and uses the following prompt: "add some background music here". The AI agent understands the context and appends this music description to the paragraph.

This approach also extends to revisions. Alice can manually change the character from a @corgi to a @cat, and Doki will regenerate the shots based on those specific changes. And Bob can use the sidebar agent for an agentic revision. He can prompt it to "adapt the story to happen in a New York City with a cat as the main character". The agent then revises the entire document, updating all definitions and shots to fit the new theme.

The examples above only demonstrate the basic usage of Doki. In real-world use, participants employ more intricate and blended workflows, which we examine in section 5.

4.2 A Structured Text Representation for Generative Videos

Document as Video, Paragraph as Sequence, Sentence as Shot. At its core, Doki adopts a three-level hierarchical structure that draws inspiration from video production: the document as a whole maps to a video, paragraphs map to sequences, and sentences map to

individual shots. This mirrors both the logic of filmmaking and the way humans naturally structure writings.

At the highest level, Doki conceptualizes the entire document as a video project. The arrangement of text determines the structure of the resulting video.

Within this framework, each paragraph corresponds to a sequence. Just as sequences in film bring together multiple shots to form a coherent scene, paragraphs gather sentences around a continuous narrative. A new paragraph often marks a shift in focus, such as transitioning from one topic to another, which parallels the cinematic logic of moving from one scene to the next.

At the most granular level, sentences maps to shots. In filmmaking, a shot is a single, uninterrupted take. In Doki, authors insert a shot preview element into the document to begin a new shot. All the sentences that follow this marker, up to the next shot or paragraph break, are treated as the textual description of that shot.

This three-level hierarchy resonates with how writers already think and compose: when beginning a new topic, they start a new paragraph; when articulating a small unit of idea, they write a sentence. Doki aligns these practices with the structure of video production.

Parametrization and Context. Beyond the basic document structure, Doki enriches this text representation through parametrization and context handling. We introduce a definition system that allows authors to define and reuse key elements consistently across the document. It also supports automatic reference resolution, ensuring continuity across sequences and shots. Implementation details of these mechanisms are elaborated in the following sections.

4.3 Doki System

The Doki interface is intentionally minimal (Figure 1). Upon opening, users see a blank document with only a few controls to learn. In the top right corner, the interface provides a status indicator, download, video player, AI agent, and a dark mode toggle. In the bottom left corner, a settings button is available. The central interaction space is the document itself.

4.3.1 Slash Menu. When users type a slash `/` (Figure 3), a menu appears that provides quick access to key creation tools. From this menu, they can start a new video shot, create reusable definitions like characters and scenes, or add audio to their project.

4.3.2 Shots. In Doki, *shot* is the fundamental unit of a visual story. Each shot is embedded directly in the text as an inline element, tightly woven into the flow of the narrative. This design allows writing and visuals to coexist in the same space.

Creating a Shot. To create a shot (Figure 4), a user types `/` → 🎬 New Shot, which inserts a shot preview node 🎞. The description that follows the node serves as the raw prompt.

Shot Sequences. As we described in subsection 4.2, paragraphs in Doki function as analogues to sequences. Users may insert multiple shots within a single paragraph (Figure 5). In this case, later shots use the image generated by the preceding shot as contextual input for the model. This supports continuity across a sequence. For example, writing:

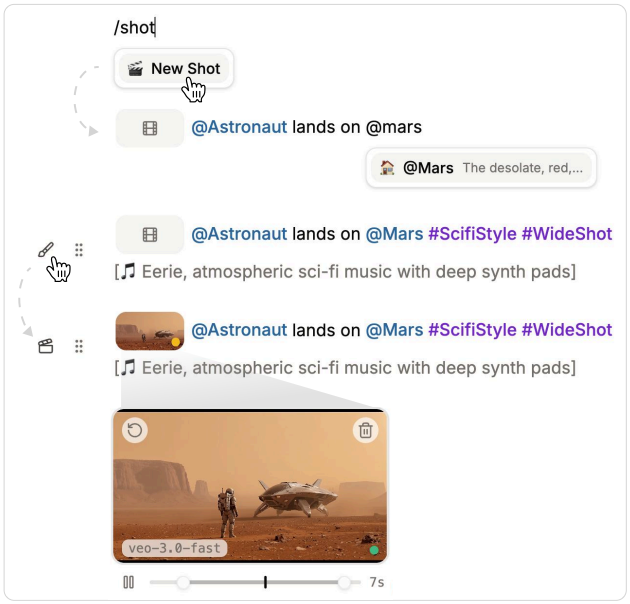

Figure 4: Creating a shot in Doki. A new shot is inserted inline with a slash command, and the description that follows serves as its prompt. The system first generates a preview image, which can then be turned into a video clip. Users can click to expand them for playback and additional controls.

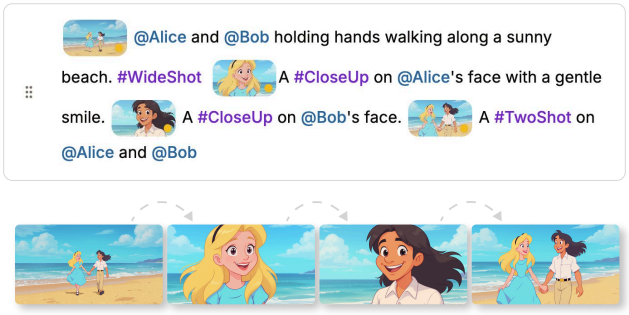

Figure 5: Writing consecutive shots within a single paragraph. Later shots inherit context from earlier ones, enabling continuity across a sequence. We achieve strong consistency between shots without repetitive context description in the Doki document.

🎞 @Pandi riding a bike. 🎞 #CloseUp on @Pandi

produces two connected shots. The second builds on the first by applying a close-up shot on the character in the same scene.

Shot Variants. Users can generate multiple variants (Figure 7) of a shot by right-clicking and selecting Add Variants. For example:

🎞 🎞 🎞 @Pandi riding a bike.

This action creates three alternative previews using the same prompt. The first is selected by default, while the others appear

semi-transparent as backups. Users can drag any variant to the front, establishing it as the active version.

Generating Image and Video. Each shot follows a staged generation pipeline: text → image → video. The system first uses the shot's description to generate a preview image, which then will be used as the first frame for video generation. The inline shot preview node will be updated accordingly as this progress proceeds. This staged pipeline offers authors control and aligns with existing creative workflows. Our formative studies revealed that creators typically prefer generating an image before producing video, as it provides an opportunity to preview visual direction before committing to a full video. Cost differences reinforce this preference: generating video costs $3.20 per clip (Veo 3), compared to $0.04 per image generation (Imagen 4). For shots in the same paragraph, Doki processes each shot in order, using prior images as contextual references for subsequent ones (details in subsubsection 4.3.5).

Shot Status. Each shot displays its current status through a small circular indicator in the bottom-right corner. The indicator communicates whether a shot is generating an image (breathing yellow), image ready (static yellow), generating a video (breathing green), video ready (static green), or outdated (red). A shot becomes outdated when its prompt or any referenced definition changes.

Expanded Shot. While shots are compact inline elements by default, they can be expanded with a click. This expanded view (Figure 4) provides additional controls, allowing users to trim videos, delete, or regenerate. Model information is also displayed.

4.3.3 Parametrized Definitions. Definitions are reusable parametrized building blocks in Doki. They allow creators to specify characters, scenes, styles, and other elements once and reference them throughout the document.

Types of Definitions. Doki supports two categories of definitions: @Mentions and #Hashtags. Mentions define elements of the story, while hashtags specify qualities that shape how the story appears. Mentions act like nouns, while hashtags act like adjectives.

@Mentions include:

- 👤 Character — a character in the story (*e.g.,* @Pandi, a little red panda wearing a suit and sunglasses).
- 🏠 Scene — a location or setting (*e.g.,* @subway, a crowded subway cabin in Tokyo).
- 🧩 Ingredient— an object or prop (*e.g.,* @sushiBox, a wooden bento box filled with sushi).
- 🖼️ Frame — a reference frame (*e.g.,* @PandiSubway, a frame of @Pandi in the @subway).

#Hashtags include:

- 🎨 Style — the overall visual aesthetic (*e.g.,* #filmnoir, #claymation, #anime).
- 🎥 Camera — cinematic movements (*e.g.,* #closeUp on @Pandi, #Pan across @TokyoTower).
- 📐 Composition — spatial arrangement or framing (*e.g.,* #twoShot, #wideShot).
- ✨ Ambience — mood, color, or lighting (*e.g.,* #sunsetGlow, #neonLights, #misty).

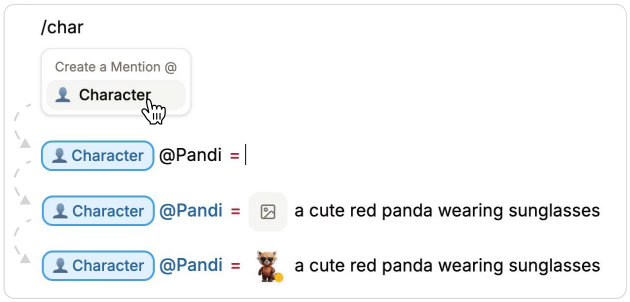

Figure 6: Creating definitions in Doki. Users open the command menu with a `/`, select a type, and provide a name and description. Optionally they can add a visual definition for even better consistency.

Doki also provides a built-in cinematography library with professional terms such as #CloseUp, #Pan, and #Dolly, making it easy to apply established film techniques.

Creating Definitions. To create a new definition, users type the slash `/` to reveal the menu, then select the definition type. This inserts a tag indicating the definition type, followed by the appropriate prefix symbol (either @ for mentions or # for hashtags). Immediately after the prefix, users first provide a name for the definition, then enter an equals sign = and a natural language description (Figure 6). For example:

👤 Character @Pandi = a little red panda

Definitions can be combined and nested. For instance, one character definition may include references to other characters, props, or styles. This allows users to incrementally build more complex concepts from simpler ones, while maintaining reusability across the document.

Visual Definitions. In addition to textual descriptions, users may attach a visual preview to any definition. To do so, insert a shot using the slash command `/` → 🎬 New Shot immediately after the equals sign (Figure 6):

👤 Character @Pandi = 🖼 a little red panda

If a visual is added, Doki uses it as the primary reference for all future generations involving that asset. If no visual is present, the system falls back to the text description. Similar to shots, visual definitions can have multiple variants (Figure 7). In addition, Doki also supports user-uploaded images for definitions. A user can upload a custom image by selecting Upload Image in the right-click context menu. The system automatically analyzes the image and generates a corresponding textual description using Gemini 2.5 Flash.

Referencing Definitions. Once a definition is created, it can be reused throughout the document. To reference a definition, the author types either @ or # followed by the name. Doki provides auto-suggestions as the user types, allowing quick selection from a list of existing definitions (Figure 4).

References are linked to their source definitions. Any change to a definition automatically propagates through the document.

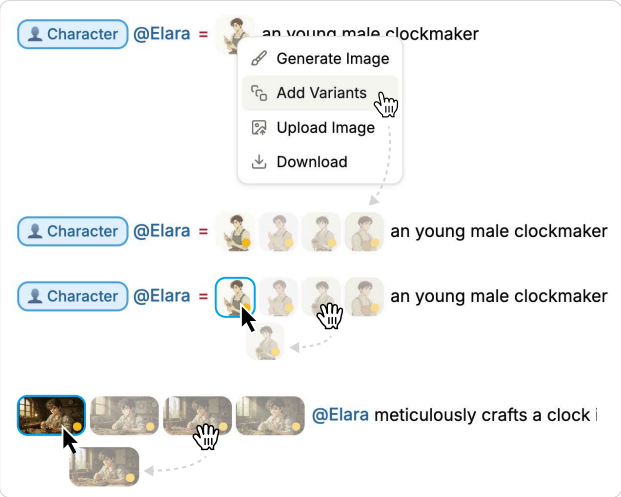

Figure 7: Adding variants to a definition and shot. The user selects Add Variants from the context menu by right-clicking on the image. Three variations will be added inline. The user can then select one variant by moving it to the first position.

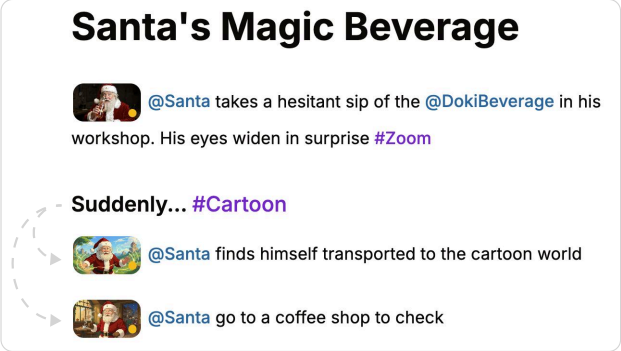

Figure 8: Authors can apply definitions globally (#all =) or scope them to specific sections via headings. In the example, a scoped definition (#Cartoon) transforms subsequent shots under the heading level into a consistent cartoon style.

For example, if @Pandi is renamed to @Panda, every occurrence of that character updates instantly. Similarly, if the description of a definition is revised or a new visual is attached, all references will reflect the updated content.

4.3.4 Global and Scoped Definitions. As documents grow in length in Doki, authors often need to apply the same definition, such as a #Style or @Scene, across multiple paragraphs. Repeating these references manually can be tedious. To address this, Doki supports both *global* and *scoped* definitions that automatically propagate without requiring explicit referencing.

To define a global attribute, users assign the name "all" to any definition. For example, writing:

🎨 Style #all = Photorealistic

automatically applies the photorealistic style to the entire document. This syntax works for all types of definitions, including mentions and hashtags.

In addition to global definitions, Doki supports heading-level scoping. Documents often contain a structured hierarchy of headings, and Doki leverages this structure to control the scope of definitions. Placing a definition on a heading applies that definition to all content within the corresponding heading level (Figure 8). This design gives authors predictable control: for example, overall stylistic rules can be set once, and specific deviations can be introduced at finer levels without redundancy.

4.3.5 Paragraph Handles. When hover over a paragraph, up to two buttons may appear on the left. On the right, the handle ⠿ allows users to reorder the paragraph or its associated shot sequence. The left-side handles communicate available generation options. If no shot is eligible for generation, the handle remains hidden. When the system detects that preview images can be produced, it displays a paintbrush icon 🖌. If the paragraph is ready for video generation, the system instead shows a clapperboard icon 🎬. Clicking the paintbrush triggers image generation for the paragraph. Clicking the clapperboard initiates video generation. The system does not regenerate all assets indiscriminately. Instead, it uses an algorithm to determine what needs to be generated and in what order, as explained in algorithm 1 in Appendix.

4.3.6 Audio. Users can add audio to their shots through the slash command /. From this menu, they may insert 🎤 Speech, 🎵 Music, or 🎧 SFX. Authoring audio is via text in square brackets. The system interprets any text enclosed in brackets as audio. For clarity, the notation appears in gray and can be mixed with normal text.

When audio notations are included, they are passed to video generation models that support synchronized audio, such as Veo 3 and Veo 3 fast. If no audio is specified, the system produces no/minimal sound effects.

4.3.7 AI Agents. Doki integrates two AI agents that provide complementary forms of assistance: the Sidebar Agent, which operates as a turn-based conversational assistant, and the Inline Agent, which supports direct, text-level edits within the editor. Both agents can perform small adjustments as well as large, multi-step edits. All edits are applied directly to the document and highlighted visually. Implementation details of the two AI agents can be found in Appendix 9.2.

Sidebar Agent. The Sidebar Agent can be opened from the main header and interacts with users through a conversational interface. Because the agent has full context of the document, it can carry out edits that span text, definitions, and generated media.

For example, it can support requests like:

- Expanding concepts. *e.g.,* "Make the #goldenHour definition more detailed."
- Inserting visual references. *e.g.,* "Add a visual reference to the @castle definition."
- Refining pacing and tone. *e.g.,* "Fix the pacing of the video. Make the first two shots more dramatic and add a climactic final shot."

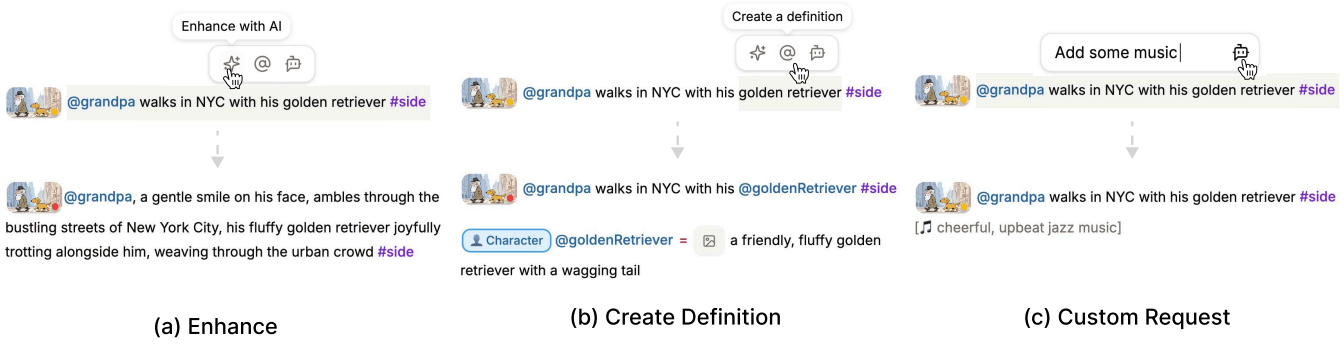

Figure 9: Three types of inline agent actions. (a) "Enhance" increases the descriptiveness of the selected text. (b) "Create Definition" converts the selection into a reusable definition. (c) "Custom Request" applies an in-context edit based on user instruction.

Inline Agent. The Inline Agent supports immediate, in-context editing within the document. It appears only when text is selected, showing a bubble menu with three options: Enhance, Create Definition, and Custom Request (Figure 9): *Enhance* increases the descriptiveness of the selected text. *Create Definition* converts the selection into a reusable definition and replaces the text with that reference. *Custom Request* opens a small input field for user instructions.

With the inline agent, authors can begin with rough drafts or loosely organized ideas and progressively refine the drafts into structured narratives. For example, a writer might draft the sentence "A young boy runs across the courtyard." At first, the boy is an incidental figure, but the author later decides to establish him as a recurring character. By selecting "A young boy" and clicking @ to turn it into a persistent @Character definition. Then, the author can issue a custom request, such as "make him wear a hat", to expand the reference to add more details.

4.3.8 Other Notable Features. Doki also provides a set of other features to support the authoring process in the header and the settings panel (Figure 12).

- *Status Indicator*: A small circular icon communicates the current state of the system.
- *Export*: Users can export their work at any time. The download menu supports three formats: the full rendered video, a zip file of all individual shots, and a JSON file containing the project's structured data. This enables seamless transfer across devices and external editing tools.
- *Video Player*: A built-in video player allows users to preview their generated content without leaving the application. The player can be toggled open or closed and provides basic playback controls for reviewing the full sequence. It also supports shot-level navigation. As playback advances from one shot to the next, the system automatically scrolls the document to the corresponding section.
- *Settings Panel*: The lower-left corner includes a settings menu for project-level actions. "Load" allows users to open a saved project from a JSON file. "New" starts a fresh project. Users may select among supported AI models for text, image, and video generation.

4.4 Doki's Shot Generation Pipeline

To transform a user's text input into a video shot, Doki maintains three distinct types of prompts throughout this process (Figure 10). The first is the *user prompt*, which corresponds to the text entered directly into the editor. The second is the *structured prompt*, which augments the user input with relevant references to definitions. The third is the *rewritten prompt*, which refines and polishes the structured prompt to optimize performance for downstream image and video generation models.

4.4.1 Structured Prompt. Once the system receives the user prompt, it parses the text and resolves references to produce a structured prompt through a tree structure. This step grounds each reference in the prompt with its corresponding definition. For example, for the user prompt: @Hedgehog driving his @bus #3D, the system transforms it into:

```
DESCRIPTION: @Hedgehog driving his @bus #3D
DEFINITIONS:
Character @Hedgehog = a cute hedgehog in a white shirt
Ingredient @bus = London red bus
Style #3D = 3D animation style
```

Instead of relying on the user prompt, Doki uses structured prompts to maintain consistency across revisions. For instance, updating the description of @Hedgehog will automatically mark all shots involving that character as "outdated" (little red dot on bottom right), even if the user did not modify the shot text directly.

4.4.2 Rewritten Prompt. Doki then refines the structured prompt through a prompt rewriter. The structured prompt is accurate but often rigid and unnatural. The rewriter addresses this by producing a fluent natural language description optimized for generative models. In practice, the rewriter is implemented as a large language model prompted with carefully designed instructions. Prompts used for image and video rewriters are available in the supplementary materials.

4.4.3 Selecting Image References. Beyond resolving textual references and refining prompts, the system also retrieves relevant visual references to guide shot generation. When a definition includes an associated preview image, the system incorporates that image as a

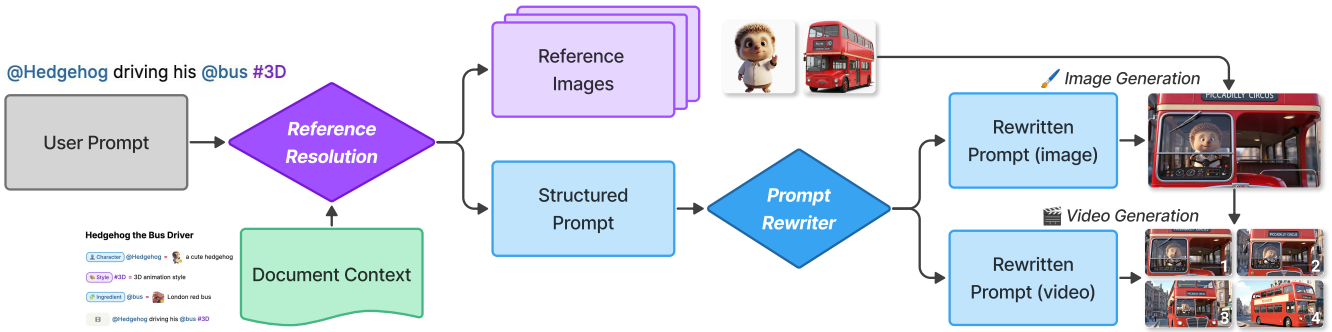

Figure 10: Doki's shot generation pipeline. First, the user's raw prompt and the document context is passed to reference resolution module to create a structured prompt and gather relevant visual reference images from the definitions. Then, the prompt rewriter rewrites and polishes this structured prompt for image and video generation. Doki first produces a static preview image and then uses that image as a the first frame to generate the final video clip.

reference. In addition, when a paragraph contains multiple shots, the model retrieves the image from preceding shot in the same paragraph. This mechanism preserves visual continuity across a sequence, as illustrated in Figure 5. In addition to rule-based strategies, the system uses a prompted LLM (full prompt in supplementary materials) to identify image references that are contextually relevant but not explicitly referred to. For instance, if a story introduces a location, shifts to another scene, and later returns to the original setting, Doki can infer that earlier images of the location should be referenced to maintain consistency.

4.4.4 Image and Video Generation. Once the system produces a rewritten prompt and retrieves reference images, it can now generate images and video for a shot in Doki. As described earlier, generation always begins with a static image preview that serves as a first frame, and then from there generate the video.

For image frame generation, the model combines the rewritten prompt with the retrieved reference images as context. By default, Doki uses the `Flux Kontext Pro` model for image generation, which takes both text and images (up to 4) as input. Video generation defaults to `Veo 3 Fast`. Doki also supports a range of different models as listed in subsection 9.3.

4.5 Implementation

Doki is built with TypeScript and React for the frontend, TipTap for rich text editing, Zustand for state management, Node.js for backend, and FFmpeg for video processing.

5 Diary Study

We conducted a week-long mixed-methods diary study of Doki. We designed our study around three research questions:

- RQ1. How do users perceive the benefits and limitations of Doki's text-native interface?
- RQ2. What workflows do people employ when using Doki?
- RQ3. How does prior experience with video editing and generative AI influence the ways they perceive and use Doki?

We chose a diary study over a lab study for three reasons. First, there is no established point of comparison for generative video authoring. Existing tools are designed for captured content and follow very different paradigms. Second, because generative video is new, most people have little prior experience, and a short session can only reveal initial impressions rather than deep insights. Third, Doki is a freeform system – we expected participants to adapt the system and develop their own workflows, which can only emerge with extended use. A diary study offered the best conditions to help us learn these evolving practices.

5.1 Participants

We recruited ten participants (6 female, 4 male; Avg. age=32.4, SD=9.9, range=[24, 57]) for a week-long diary study of Doki. Six were recruited from outside of our organization and four from within. Roles included product designers (2), animators (2), a filmmaker, a graphic designer, a software engineer, a content creator, a UX designer, and a program manager (Table 2).

Participants reported different levels of experience and frequencies of video creation: 4 created videos daily, 3 weekly, and 3 a few times per year. Their videos spanned diverse contexts, such as professional films, social media content (*e.g.*, TikTok, Reels, Shorts), academic/work projects, and personal vlogs. Participants used multiple video editing tools. Adobe Premiere Pro was most common (8/10), followed by iMovie (5) and CapCut (5). Three reported using specialized generative video platforms (3/10, e.g., Runway, Pika, Google Flow).

Participants' experience with generative AI tools also varied. For large language models, 5 reported daily use while 3 shared rarely using them. Image and video generation tools were less commonly used: 4 participants reported using image generation models at least once per week, and 2 reported using video generation models at least once per week. Notably, the participants who used video generation models regularly had professional roles closely tied to generative video creation: one is a filmmaker producing AI-driven video content; the other a product designer responsible for producing animation videos for their brand. Overall, seven participants were already using generative AI in their existing video creation workflows. For example, scriptwriting with ChatGPT, storyboard ideation with Midjourney, refining edits with Photoshop's AI features, etc. 3 participants reported minimal or no prior experience with AI tools – providing contrast in perspectives and practices.

5.2 Procedure

The study followed a three-phase structure:

Phase 1: Onboarding (60 minutes). Participants completed informed consent and a background questionnaire, received a guided tutorial to Doki's core concepts (e.g., shots, definitions, slash menu, agents) and completed a short open-ended creative task ("*Create a 30-second video about anything*") with support from one of the authors.

Phase 2: In-the-Wild Diary Study (5 days). Participants used Doki independently for five days. Each day, they were instructed to engage with the tool for at least 50 minutes, with the goal of producing 2-3 complete videos at the end of 5 days. Participants were encouraged to follow their natural workflow, allowing ideation, drafting, generation, and iteration to unfold across days.

After each day, participants completed a brief survey. To capture an overall measure of participant satisfaction, we included a single-item rating question (1-10): "*Please rate your overall satisfaction with your Doki experience today*". Participants also reported their workflow strategies, identified moments of ease and difficulty, and listed features used (*e.g.,* shot generation, definitions, AI agents, etc.). Each survey concluded with an optional submission of the day's project file and exported video.

In addition, we collected interaction logs spanning session duration; number of shots created or deleted; editing operations; words typed; use of mentions and hashtags, previews, and menus; generative activity (*e.g.,* attempted/successful image or video generations, regeneration counts, total generation time); export behavior; AI assistant interactions; and cost metrics based on image and video generations.

Phase 3: Exit Interview (60 minutes). Participants completed a semi-structured remote interview. The first portion consisted of a retrospective think-aloud session [16], in which participants reviewed one of their projects, explaining creative decisions, feature use, and how the final video evolved. We also asked about overall experience, workflows, ownership, and learning. Participants also completed the System Usability Scale (SUS) [5] to provide standardized quantitative feedback.

Apparatus. Participants accessed Doki remotely as a web app during the week-long study period. External participants were compensated $30/hour. Collected data included submitted videos, Doki documents, surveys, interaction logs, and interview recordings/-transcripts.

6 Results

6.1 Doki Usage

6.1.1 Interface Usage. Excluding the onboarding and exit interviews, participants spent an average of 91.7 minutes per session (SD = 45.5; range: 40–217) editing with Doki, indicating engagement beyond the required 1 hour per day. On average, they generated 45.5 images (SD = 33.2) and 20.3 videos per session (SD = 15.8). Normalized by time, participants produced approximately 0.6 images and 0.3 videos per minute (Figure 13).

The most frequently used features were export (96% of sessions), shot generation (90%), and the video player (90%). Participants also made extensive use of defining hashtags (82%) and mentions (78%). Shot variants were generated in 76% of sessions, and audio was added in 68%. The chat sidebar AI agent was used in 74% of the sessions while the inline agent was used 50%. Heading-based styles were the least used feature, appearing in only 24% of sessions.

6.1.2 What Videos Did Participants Create with Doki? Participants submitted 46 video files in total. The average duration was 67.14 seconds (SD = 35.97; range: 15.04–184.04). Most videos (31) fell within 30-90 seconds; Four were shorter than 30 seconds, and eleven exceeded 90 seconds.

Participants produced a variety of videos (Figure 11) that fell into five main categories: storytelling, instructional video, advertising, music, and experimental. Storytelling was the most common, with eight participants (P2, P3, P4, P5, P7, P8, P9, P10) creating original narratives such as fairy tales, pet stories, or character-driven animations. One participant (P1) focused on instructional formats, producing cooking videos and academic explainers. Advertising and promotional content appeared in three cases: P2 created two claymation-style advertisement for Doki, P6 worked on a summer camp promotion video, and P8 produced a series of animal welfare videos tied to World Dog Day. Some participants explored less conventional directions. P8 experimented with music videos. Two participants also created experimental projects like a surreal animal protest (P2) and ASMR (P7).

While many participants used Doki for narrative storytelling, these projects demonstrate the breadth of video types Doki can support. We include selected videos authored by participants in the supplementary material.

6.2 RQ1. How do users perceive the benefits and limitations of Doki's text-native interface?

6.2.1 Overall impression of Doki. Overall, participants reported a positive experience with Doki. The System Usability Scale (SUS) ratings (Table 1) collected during the exit interviews averaged 81.2 (SD = 8.9, median = 83.8). This score corresponds to the *Excellent* category defined by Bangor et al. [3], placing Doki within the 90–95th percentile (benchmark average = 68). Participant ratings ranged from a 62.5 to 90.0, with all but one participant rating Doki at or above 72.5. These findings are consistent with daily survey measures of overall satisfaction, which had a mean of 7.48/10 (SD = 1.68, median = 8.0).

Beyond the quantitative measures, many participants described Doki as a robust system that generates high-quality visuals and is delightful to use. For example, P8 shared that "*This whole process makes me feel relaxed and gives me a great sense of accomplishment.*" Similarly, P1 remarked, "*Doki is a really great generative AI video tool, probably one of the best I've used so far.*"

6.2.2 Ease of use and learnability. Nine out of ten participants described Doki as very easy to use, highlighting its simple interface. They noted that the system required little effort to learn and that they could quickly grasp its core functionality. For example, P3 and P5 reported that common features such as the slash command, mention system, and hashtags mirrored those in platforms they already used daily. This familiarity helped them learn quickly and made the features easy to remember. P8 found Doki much simpler than

Figure 11: Example frames from videos created by participants in our diary study. The videos demonstrate the breadth of video types Doki supports. Each row corresponds to a single video, shown left to right through four representative keyframes.

Id	Role	Video Creation Freq.	GenAI Use (Text)	GenAI Use (Image)	GenAI Use (Video)	SUS
1	UX Designer	A few times a year	A few times a year	Monthly	A few times a year	87.5
2	Filmmaker	Daily	Daily	Daily	Daily	77.5
3	Product Designer	Weekly	Rarely or never	Rarely or never	Rarely or never	75.0
4	Software Engineer	A few times a year	Daily	Weekly	Monthly	90.0
5	Animator	Daily	Rarely or never	Rarely or never	Rarely or never	80.0
6	Program Manager	A few times a year	Daily	Monthly	A few times a year	72.5
7	Animator	Daily	Weekly	A few times a year	Rarely or never	62.5
8	Product Designer	Daily	Daily	Weekly	Weekly	87.5
9	Graphic Designer	Weekly	Monthly	Weekly	A few times a year	90.0
10	Content Creator	Weekly	Daily	A few times a year	Rarely or never	87.5

Table 1: System Usability Scale (SUS) scores and subscales (Usability and Learnability) across participant roles, video creation frequency, and generative AI tool use.

her usual video creation workflow because it minimized context switches:

> "*Doki combines them together so I don't need to shift between different windows...it really saves me the window shifting time.*"

These qualitative findings align with the survey results. Quantitative analysis showed no significant change in participants' satisfaction scores across the study period (Repeated Measures ANOVA, p = 0.42). Participants consistently rated their experience positively from the start. As P4 noted, "*Almost immediately... I felt like I could use it right away.*" In the exit interviews, the System Usability Scale (SUS) subscale for learnability [22] yielded a median score of 81.2, which falls within the excellent range.

Because of Doki's simple interface, three participants mentioned that Doki made them *focus on storytelling*. Compared to tools like Flash or After Effects, P9 described Doki as "*10 times easier*," and he could focus on the idea and content rather than symbol animation or movement. As P5 put it, "*I think the interface is perfect...the whole interface is very simple and clean. So I can focused on creating my story.*"

6.2.3 Fast to go from idea to content. Eight participants emphasized how Doki accelerated the process of moving from a rough idea to deliverable content. It enables participants to bypass many of the slow, manual steps in traditional workflows. P1, a former professional documentary filmmaker, contrasted Doki with traditional production workflows:

> "*For like a quick project, I think with my previous workflow that's just not possible because I have to go out to shoot, I have to spend a lot of time on editing. But with Doki I can just easily prompt the system to generate a very great first draft.*"

Several participants mentioned that this efficiency was especially valuable for fast, lightweight projects where perfect visual fidelity was not the priority. P5 reported: "*So on average it will be 15 minutes to create about one minute or longer video.*" Similarly, during the diary study, P8 produced a series of five 30-second videos for World Dog Day, in roughly ten minutes each – an output that would have been unthinkable in their prior workflow. Participants consistently noted that integrating an AI agent within the document interface was essential to this speed. As P2 described, "*It can create short level breakdowns. It can auto format the whole scenario. It can do anything for you.*"

6.2.4 Document representation fosters comprehension. Four participants explained how Doki's document-based representation not only accelerated content creation but also enhanced their comprehension of narrative flow and structure. Unlike traditional nonlinear editors (NLEs), which primarily rely on timelines, Doki offered a holistic, text-based overview of their projects. This shift was seen as particularly valuable in helping participants visualize their films in a semantic way. P2, a professional filmmaker, highlighted how this representation provided an immediate sense of coherence across the film:

> "*Doki sets itself apart in how it gives you a sense of how your film will look. You can see the flow of your scenes, how your film is looking scene by scene and shot by shot.*"

By contrast, P3 noted that conventional NLEs often lack such an outline, making it harder to conceptualize story development beyond technical editing: "*It (NLEs) doesn't have a nice visual outline for writing or editing your videos or cropping.*" Similarly, P6 explained that the document structure enhanced her understanding of video composition: "*I feel like it was helpful for me to understand the structure of [the] doc. It's pretty intuitive.*"

In addition, participants mentioned that the text representation also made it very easy for them to view and understand AI's changes, and even for them to learn how to better use Doki (P4):

> "*The notebook interface with text as the driving element is clever. You can learn from how it generates music or voice pieces, even if you wouldn't use it directly.*"

6.2.5 Parametrization makes story building easier. Six participants emphasized how Doki's parametrization design helped them build their stories. For example, by defining reusable elements such as characters, scenes, and styles, they could begin projects with a strong foundation even when the narrative was not yet fully formed. As P6 explained, "*Building reusable characters and ingredients really made sense as a lot of times I have some clarity into what/who I want to use but not the story itself.*"

Participants also described how these definitions supported coherence across their work. P10 highlighted that parametrization reduced randomness in generative outputs. "*I think this tool is really cool because you can define a character and scene so it would not be like very randomly generating the video.*" Similarly, P1 noted that Doki itself is not just a video tool but also a really good tool for prompt engineering and crafting the visuals, even just a single shot. He described that the definition system make it easier for him to build up complex scenes from smaller components:

> "*I don't have to write repeated keywords just to keep this consistent. I can just use tags and add each character so and they will stay.*"

He further envisioned that in the future, professional creators could design and share parametrized styles or camera movements as creative assets, and make them available through simple tags for others to use.

Finally, participants appreciated how parametrization integrated naturally into Doki's text-native representation. P5 noted how features like mention, hashtag and audio functioned seamlessly within the document: "*...mention and hashtag and create audio. It's all in the same logic thing. It's very easy to remember and learn.*"

6.2.6 Precise control is limited. Nine participants noted that Doki's outputs often failed to match their prompts, requiring repeated regeneration to achieve acceptable results. P1 described persistent artifacts: "*A lot of videos include nonsensical text that I couldn't remove, even when I explicitly prompted it.*" Such errors forced participants into trial-and-error workflows. As P10 explained, "*I need to regenerate image again and again...* " P8 also noted visual glitches such as "*dogs disappearing mid-run and people brushing the air instead of the animals.*" Several expressed frustration that the text-only interface constrained their ability to realize specific visual goals. P9 noted, "*Doki is not good at when you want specific frame compositions.*" This challenge was especially pronounced for users with *strong visual references*. During the study, P10 attempted to create a video from her storybook. Because she already had a clear mental image of the visuals and compositions, she found it difficult to reproduce them through text prompts. She explained, "*If you want to create your personal story, sometimes the video could not reflect everything that you want. Maybe just achieve 80% of your imagination.*"

6.2.7 Hard to work with audio and music. Participants noted constraints in how Doki handles time-based media. Because the system is organized around a paragraph and document structure, it is difficult to add audio that spans multiple paragraphs or begins asynchronously. P8, who attempted a music video, found it hard to align visuals with audio.

6.3 RQ2. What workflows do people employ when using Doki?

6.3.1 A lot of variety in how people start. Participants entered Doki from different starting points, reflecting varied creative practices. Some began with a complete script that they had written beforehand. For example, P1 noted: "*I already have a text version of that recipe, so I just use the chatbot, paste the whole recipe and ask [it] to generate all the ingredients I may need.*"

Others would first defining characters, scenes, or other assets before turning to narrative construction. As P3 explained:

> "*I'd probably define all the assets and the scenes. Then once I did that, I would tell Doki the story I want to tell and have it fill in the rest, like the story outline for me.*"

A third group described a more exploratory and freeform approach. For these participants, Doki supported writing and discovery without preparation. P4 mentioned that he can "*just begin writing something*." Similarly, P2 worked in small pieces and refined step by step:

> "*I worked on small pieces iteratively, building scenes and refining character interactions step by step.*"

Across these practices, Doki enabled participants to begin video authoring in ways that matched their own practices. Whether starting from a script, from defined assets, or from improvisational writing, participants found that the system supported flexible entry points into the creative process.

6.3.2 High reliance on AI. Eight participants reported that they would begin by using AI to generate a first textual draft before refining further. Six participants went further, relying almost entirely on Doki's AI agents throughout the process. They described starting with a rough, often one-line idea, using the AI as a jump start, and then remaining in the conversation panel or inline agent to complete their projects with minimal manual edits. P8 noted that she often preferred asking the AI to revise drafts rather than editing them manually. As she explained, "*AI knows better how to talk to AI.*" P10 described a similar reliance on delegation: "*I don't really read the script the agent creates. I just want to see how the image would look...I generate those images and then see if the agent misunderstood my description.*"

This pattern was not limited to casual creators. P2, a professional filmmaker who produced some of the best videos in our study, revealed that nearly the entire process was conducted with the AI: "*"I had just one line idea...and it really helped me to brainstorm much more in terms of how different scenes will be, how we can create different shots...and then we worked together to turn [it] into a full script.*"

Four participants described a gradual shift from manual work to heavier AI reliance as they grew more comfortable and found it reliable and easy to use. P4 explained:

> "*The workflows are slightly different. At first, I did that very manually. The second one was more hybrid type, and then I'll move to AI more and more, for the last video, I'll just type the command, tell AI what to do rather than manually edit.*"

We interpret this increasing reliance not only as a reflection of participants' trust in the system, but also as a side effect of the

design of Doki itself. By making collaboration with AI extremely simple through a text representation, participants can find it natural to delegate more and more of the process to AI. In other words, simplicity of interaction could encourage reliance.

6.3.3 Users who rely almost entirely on AI still feel ownership. An interesting finding is that, using Doki, even participants who relied almost completely on AI still felt *strong sense of ownership* of their videos. P4 cited Andy Warhol's saying:

> "*Art is the process of an artist selecting.*"

P1 similarly explained: "*Because I get to decide what every item looks like, I feel like this is my creation. That's what gives me ownership.*" Participants often compared themselves to directors, as P6 described that she feels her role was no different from a director's, except working with an AI agent instead of human agents.

Several participants also contrasted this experience with other platforms. P10 noted, "*When I use MidJourney, if other people use similar prompts, they would get 90% similar images. I would think this is generated by MidJourney, so I give the credit to MidJourney. With Doki, I feel ownership. Every choice of image or cut made me feel I'm a video maker.*" Participants found that Doki enabled a workflow where creators could delegate most production to AI while still experiencing a strong sense of authorship.

Two participants did not use AI at all. Both were animators who typically worked by hand and had little prior experience with text-to-image or text-to-video models. They expressed a preference for full creative control: "*I just want to create the story by myself.*" (P5).

6.4 RQ3. How do users' prior experience with video editing and generative AI influence the ways they perceive and use Doki?

6.4.1 Doki empowers participants without previous experience to create video stories. For participants without training in filmmaking, animation, or generative AI, Doki opened up new possibilities that had previously felt out of reach. Four participants noted that they would never have created a video story without Doki, and now, they could express ideas in new ways. For example, P10, who had no prior drawing or animation experience, described:

> "*Like just use this and generate some animation. I do not have experience drawing those, but I can use this tool to create that.*"

Others emphasized Doki's role in unlocking creativity and enabling personal expression. P9 called it "*my dream tool*," explaining that it helped deliver messages and bring childhood stories to life. For many participants, Doki meant more than just a tool – it represented a shift in what creative expression could mean for them. As P1 reflected:

> "*It kind of [offers] a new capability I didn't have before. In the past, if I want to share something with my friends, I have to write it down, but now I can create a video to express whatever I want.*"

6.4.2 Experts position Doki as complementary rather than substitutive. Participants with professional experience in video editing described Doki as fundamentally different from their existing tools. Rather than viewing it as a replacement, they see it as complementary for projects outside their primary workflows. P1 explained,

> "*I think creating a video with Doki is very different than my previous video editing experience. I would even define them as two completely different tasks, and are for very different purposes.*"

Others emphasized this difference by situating Doki within the production pipeline. P4 viewed traditional editing tools to be essential for precision and polish, while Doki was valued for its speed in ideation and draft generation. Similarly P5 stated:

> "*I think I will use [Doki in] the first period of my animation making process because I can first have a script and then input it into Doki to create a video for me to as a reference.*"

Yet even as professionals stressed that Doki could not substitute for high-fidelity outputs, they also highlighted the dramatic contrast in efficiency. P5, who normally works through labor-intensive frame-by-frame animation, reflected that producing one minute of video by hand could take two months, while with Doki she could create it in about an hour. For her, the speed advantage was undeniable, even though the resulting output still carried a "*clearly AI feeling.*"

Another recurring challenge for professionals was that their existing expertise felt difficult to transfer into Doki. P10 reported, "*Although I learned some filming technique, I cannot use it here. Even if I have some idea in my mind, I don't know how to describe in words.*" Similarly, P7 expressed frustration that drawing skills could not be applied directly. For professional with high quality standards, the visual quality Doki can generate still falls short:

> "Animation industry standard is way higher for now…for animators, this is far from enough." – P7

6.4.3 Prior experience with generative tools leads to greater satisfaction. Participants with previous experience in generative AI video tools reported greater appreciation for Doki's capabilities. P1, who had recently worked on a generative video research project, explained:

> "*I tried a lot of different tools, including Google's Veo 3, Google Flow, and a tool that we developed, [but] the consistency that I can generate across the scenes is just uncanny. Doki did something that was not previously possible.*"

Similarly, P2 noted that long-term familiarity with generative tools gave him a clearer sense of their strengths and limitations, a perspective that new users may lack:

> "*Now that I have been using these tools for more than two years, I still don't have an idea how good they are with each of these styles…new users would have no idea.*"

Film knowledge also emerged as an important factor. Participants with professional or semi-professional backgrounds were more adept at structuring stories and translating them into coherent visual sequences. As P2 explained,

> "*It usually requires a craftsmanship which most people don't have. People do have ideas, but people don't know*

how to create an entire piece. That's where expertise comes in."

7 Discussion and Future Work

While text-native authoring made it easier for participants to get from idea to video, we found that a low-barrier to creation does not on its own result in high-quality storytelling.

Approachability does not guarantee quality. A text-native interface lowers barriers, but strong stories still demand craft. In our diary study, participants moved quickly from concepts to completed videos, yet not all videos had compelling arcs. Doki's document-centric structure makes working with narrative elements like characters and scenes explicit and editable. Expert film makers know how to structure a narrative and how to sequence shots and scenes, while novices are left unsure how to improve their video. Future work includes optional narrative scaffolds (*e.g.,* three-act templates, A/B/C plots) and lightweight diagnostics (pacing irregularities, under-specified protagonists) that help authors see and improve a story without adding UI complexity.

Do we really need a video model for longer videos? Current text-to-video systems cap out at short durations (5–10 seconds[15, 29]). At first glance, this seems like the primary limitation. Yet the average shot duration in contemporary film is roughly 4 seconds [11], well within today model's capability. Our formative and diary studies suggest that the main gaps lie elsewhere: *consistency* (stable characters, props, styles), *control* (camera grammar, timing, transitions), and *context* (memory across shots and sequences). In Doki, we treat duration as a compositional rather than monolithic model problem. Text generates *keyframes* that anchor identity, layout, and style; these keyframes expand into short, controllable shots. Longer models will still matter for specific cases (e.g., "oners," dance, sports, talking heads), but for many scenarios, progress on cross-shot memory, identity preservation, and story-aware control is likely to deliver more value than raw clip length.

Document as a human–AI common ground. Most current AI creation tools such as Lovable (text-to-website), Deep Research (text-to-report), or Eleven Labs (text-to-audio), follow a direct input–output model: a prompt produces a finished artifact with limited opportunities for inspection or revision along the way. By contrast, Doki introduces the document as an intermediate representation: a format that is simultaneously readable and editable by humans and interpretable and executable by AI.

In future iterations, even if Doki begins from a simple prompt box, the output would not be a locked timeline or rendered video, but rather a document representation: a shared working space between human and AI where operations remain legible, revisable, and easy for user intervention.

Doki's document representation approach highlights a broader opportunity for the HCI community: how to design effective *intermediate spaces* that make AI activity transparent and at the same time create richer opportunities for human agency.

Temporal expressivity limits of a document representation. A linear document is limited in expressing any temporal concurrency, cross-cut constructs, and duration-sensitive rhythm. Participants struggled to specify overlapping or offset audio (e.g., pre-lapped dialogue, music that bridges sequences), and transitions that bind across shot boundaries (e.g., J/L cuts, cross-dissolves, match cuts) without resorting to external tools. We implemented inline audio notations (§4.3.6) to address basic needs, but finer control remains awkward. One path forward is to extend the Doki language with lightweight temporal primitives that preserve readability while increasing control: inline event offsets (e.g., `[SFX door slam +1.2s]` or `[MUSIC fade in 4s]`), paragraph- or sequence-scoped beat/tempo markers (e.g., `#Tempo=110 BPM`), and parameterized transition definitions that bind explicitly to adjacent shots (e.g., `#CrossDissolve 12f`, `#JCut`).

Compositional structures vs. single representation. Most video authoring tools rely on "compositional structures" with interconnected underlying data. Professional non-linear editors such as Premiere Pro and Final Cut present projects as a network of linked views: an asset library, a multi-track timeline, a source monitor, and often a transcript panel for editing by text. Descript [12] aligns transcript editing with a timeline. Recent research systems adopt the same principle. VideOrigami [7] links scripts, storyboards, canvases, and timelines.

Doki explores a radically different direction. Instead of coordinating across multiple views, it centralizes all representations within a single text-native substrate. Authoring video is reframed as writing. This presents a different set of tradeoffs. Multiple views can indeed support task-specific efficiency, such as precise pacing control in a timeline or spatial layout on a canvas. But Doki, for a lot of people whether they are professionals or amateurs, greatly simplifies the interface and interaction: there are very few mechanisms to learn and people can begin authoring almost immediately. This unified interface also strengthens version control and collaboration between humans or AI, while introducing challenges in temporal limits and etc. From an HCI perspective, these paradigms are not in opposition but explore distinct possibilities. In a sense, they embody two different philosophies: compositional structures seek to optimize every step of authoring with specialized views, while Doki leverages AI to pursue an ultimately minimalistic interface.

8 Conclusion

In this paper, we introduced Doki, a text-native interface for authoring generative videos. Findings from a week-long diary study show that it enables faster idea-to-content workflows, improved coherence via parameterization, and clearer narrative comprehension compared to traditional tools.

More broadly, we believe that Doki contributes a paradigm that positions text not only as input to generative systems but as the primary substrate for narrative, structure, and production. We believe that as generative models continue to expand in capability, it is essential to rethink interface paradigms for content creation. Doki demonstrates one such direction – a low-barrier, text-native model of authoring – and opens new questions about how natural language might serve as the foundation for future creative tools.

References

[1] PJ Ace. 2025. *PJ Ace YouTube Channel.* https://www.youtube.com/@pjacefilms
[2] Acqualia Software OÜ. 2025. Soulver – Natural-Language Notepad Calculator. https://soulver.app. Accessed: 2025-09-04.

[3] Aaron Bangor, Philip T Kortum, and James T Miller. 2008. An empirical evaluation of the system usability scale. *Intl. Journal of Human–Computer Interaction* 24, 6 (2008), 574–594.

[4] Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Kyle Lacey, Yam Levi, Li Cheng, Dominik Lorenz, Jonas Müller, Dustin Podell, Robin Rombach, Harry Saini, Axel Sauer, and Luke Smith. 2025. FLUX.1 Kontext: Flow Matching for In-Context Image Generation and Editing in Latent Space. *arXiv preprint* arXiv:2506.15742v2 (June 2025). https://arxiv.org/abs/2506.15742v2

[5] John Brooke et al. 1996. SUS-A quick and dirty usability scale. *Usability evaluation in industry* 189, 194 (1996), 4–7.

[6] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, et al. 2024. Video generation models as world simulators. *OpenAI Blog* 1, 8 (2024), 1.

[7] Yining Cao, Yiyi Huang, Anh Truong, Hijung Valentina Shin, and Haijun Xia. 2025. Compositional Structures as Substrates for Human-AI Co-creation Environment: A Design Approach and A Case Study. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. 1–25.

[8] Paul Chandler and John Sweller. 1991. Cognitive load theory and the format of instruction. *Cognition and instruction* 8, 4 (1991), 293–332.

[9] Paul Chandler and John Sweller. 1992. The Split-Attention Effect as a Factor in the Design of Instruction. *British Journal of Educational Psychology* 62, 2 (1992), 233–246. https://core.ac.uk/download/pdf/194661366.pdf

[10] Peggy Chi, Tao Dong, Christian Frueh, Brian Colonna, Vivek Kwatra, and Irfan Essa. 2022. Synthesis-Assisted Video Prototyping From a Document. In *UIST '22: Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*. https://doi.org/10.1145/3526113.3545676

[11] James E Cutting. 2016. The evolution of pace in popular movies. *Cognitive research: principles and implications* 1, 1 (2016), 30.

[12] Descript, Inc. 2025. Descript. https://www.descript.com/.

[13] Alisa Fortin, Guillaume Vernade, Kat Kampf, and Ammaar Reshi. 2025. *Introducing Gemini 2.5 Flash Image (aka "nano-banana")*. Technical Report. Google AI / Vertex AI. https://developers.googleblog.com/en/introducing-gemini-2-5-flash-image/

[14] Ohad Fried, Ayush Tewari, Michael Zollhöfer, Adam Finkelstein, Eli Shechtman, Dan B. Goldman, Kyle Genova, Zeyu Jin, Christian Theobalt, and Maneesh Agrawala. 2019. Text-based Editing of Talking-head Video. *ACM Transactions on Graphics* 38, 4 (2019), 68:1–68:14. https://doi.org/10.1145/3306346.3323028

[15] Google DeepMind. 2025. Veo 3. Web page. https://deepmind.google/models/veo/ State-of-the-art text-to-video generation model capable of producing synchronized audio—including dialogue, sound effects, and ambient noise—with high realism and prompt adherence.

[16] Zhiwei Guan, Shirley Lee, Elisabeth Cuddihy, and Judith Ramey. 2006. The validity of the stimulated retrospective think-aloud method as measured by eye tracking. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*. 1253–1262.

[17] Mina Huh, Saelyne Yang, Yi-Hao Peng, Xiang'Anthony' Chen, Young-Ho Kim, and Amy Pavel. 2023. Avscript: Accessible video editing with audio-visual scripts. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–17.

[18] Ink & Switch. 2025. Embark – Dynamic Documents for Making Plans. https://www.inkandswitch.com/embark/. Accessed: 2025-09-04.

[19] Ink & Switch. 2025. Potluck – Dynamic Documents as Personal Software. https://www.inkandswitch.com/potluck/. Accessed: 2025-09-04.

[20] Dan Kieft. 2025. *Dan Kieft YouTube Channel*. https://www.youtube.com/@Dankieft

[21] Kling AI. 2025. Kling AI: AI Image & Video Maker. https://www.klingai.com/.

[22] James R. Lewis and Jeff Sauro. 2009. The Factor Structure of the System Usability Scale. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Vol. 53. SAGE Publications, 1259–1263. https://doi.org/10.1177/154193120905301905

[23] Jing Li, Yang Chen, Rui Zhao, et al. 2025. PromptCanvas: Composable Prompting Workspaces Using Dynamic Widgets. *arXiv preprint arXiv:2506.03741* (2025). https://arxiv.org/abs/2506.03741

[24] Luma Labs. 2025. Luma AI – 3D Capture and Generative AI. https://lumalabs.ai.

[25] Notion Labs Inc. 2025. Notion. https://www.notion.com/product/docs.

[26] OpenAI. 2024. *Introducing Canvas in ChatGPT*. https://openai.com/index/introducing-canvas/

[27] Pika. 2025. Pika Labs – AI Video Generation. https://pika.art.

[28] Steve Rubin, Floraine Berthouzoz, Gautham J Mysore, Wilmot Li, and Maneesh Agrawala. 2013. Content-based tools for editing audio stories. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*. 113–122.

[29] Runway. 2025. Runway – AI Magic Tools for Creators. https://runwayml.com.

[30] AI Video School. 2025. *AI Video School YouTube Channel*. https://www.youtube.com/@aivideoschool

[31] Kevin Stratvert. 2025. *Kevin Stratvert YouTube Channel*. https://www.youtube.com/@KevinStratvert

[32] Bekzat Tilekbay, Saelyne Yang, Michal Lewkowicz, Alex Suryapranata, and Juho Kim. 2024. ExpressEdit: Video Editing with Natural Language and Sketching. arXiv:2403.17693 (ACM IUI 2024). https://doi.org/10.48550/arXiv.2403.17693

[33] Anh Truong, Floraine Berthouzoz, Wilmot Li, and Maneesh Agrawala. 2016. QuickCut: An Interactive Tool for Editing Narrated Video. In *UIST '16: Proceedings of the 29th Annual ACM Symposium on User Interface Software and Technology*. 497–507. https://doi.org/10.1145/2984511.2984569

[34] Bret Victor. 2006. Explorable Explanations. In *Proceedings of the ACM Conference on User Interface Design*. https://worrydream.com/ExplorableExplanations/#reactiveDocument

[35] Bryan Wang, Yuliang Li, Zhaoyang Lv, Haijun Xia, Yan Xu, and Raj Sodhi. 2024. Lave: Llm-powered agent assistance and language augmentation for video editing. In *Proceedings of the 29th International Conference on Intelligent User Interfaces*. 699–714.

[36] Miao Wang, Guo-Wei Yang, Shi-Min Hu, Shing-Tung Yau, and Ariel Shamir. 2019. Write-A-Video: Computational Video Montage from Themed Text. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)* 38, 6, Article 177 (2019). https://doi.org/10.1145/3355089.3356520

[37] Haijun Xia. 2020. Crosspower: Bridging Graphics and Linguistics. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology (UIST '20)*. https://doi.org/10.1145/3379337.3415845

[38] Haijun Xia, Jennifer Jacobs, and Maneesh Agrawala. 2020. Crosscast: Adding Visuals to Audio Travel Podcasts. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology (UIST '20)*. https://experts.colorado.edu/display/pubid_385971 Short paper.

9 Appendix

9.1 Technical Details

9.1.1 JSON example. We show a simplified example JSON file of Doki to illustrate its underlying data structure:

```
{
   "shots" : [
     {
        "id" : "shot-1757235383722-pynguler9",
        "context" : "definition",
         "structuredPrompt"  : "CHARACTER: corgi, a cute corgi
             with golden-brown and white fur.",
        "status" : "video-ready"
     },
     {
        "id" : "shot-1757234503129-4esljq3y6",
        "context" : "definition",
         "structuredPrompt"  : "LOCATION: airport, a small
            regional airport terminal.",
        "status" : "image-ready"
     },
     {
        "id" : "shot-1757235144859-7k10hdcyn",
        "context" : "paragraph",
         "structuredPrompt"  : "DESCRIPTION: corgi arrives at
            the airport with luggage.",
        "status" : "image-ready"
     },
     {
        "id" : "shot-1757234758231-ldfx5tr7d",
        "context" : "paragraph",
         "structuredPrompt"  : "DESCRIPTION: corgi then boards
             the plane.",
        "status" : "image-ready"
     }
   ],
   "tags" : {
      "mentions"  : [
       {  "name" : "corgi",  "tagName" : "Character" },
       {  "name" : "airport",  "tagName" : "Scene" }
     ],
      "hashtags"  : [
       {  "name" : "all",  "tagName" : "Style",   "userPrompt"  :
             "Japanese anime style" }
     ]
   }
}
```

9.2 Agentic Editing

Doki supports agentic editing that allows both the sidebar and inline agents to directly manipulate the document. This enables authors to interactively revise content through natural language commands. Here, we detail the implementation of the JSON-based editing API.

9.2.1 Document Context and Agent Memory. Both agents in Doki have access to a structured representation of the full document. This context includes definitions, headings, paragraphs, shots, and associated metadata. The sidebar agent also maintains a conversational memory that supports multi-turn interactions.

9.2.2 Editing API. Doki's AI agent editing feature is built around a structured API. All edits – whether triggered through chat or inline interaction – are represented as typed JSON objects. Each specifies:

- *id*: a unique identifier for the edit operation
- *target*: the type and id for the document node to be modified (*e.g.,* definition, shot, paragraph, heading)
- *newContent*: the replacement content (or null for deletions)
- *description*: a concise explanation of the edit that will be shown to the user

This JSON format enables a wide range of editing operations such as updating definitions, revising shots, replacing paragraphs, modifying headings, inserting or deleting content, and performing a sequence of multiple edits.

For example user asks, *"Add an establishing shot right after the introduction."* The agent translates this request into an insertion edit at the appropriate location. The JSON representation specifies the insertion point (after the introduction paragraph) and the new content to be added:

```
{
  "id": "edit-5",
   "target" : {  "type" : "insert",  "selector" : {  "after"
       : "paragraph_intro" } },
    "newContent"  : "@Hero enters the castle gates #
       WideShot",
   "description"  : "Inserted a new establishing shot after
       the introduction paragraph"
}
```

Here, we show another example of how we implemented the Create Definition feature for inline agent (Figure 9). When the user highlight the text *"a dark sorcerer with glowing red eyes"* and click `@` in the bubble menu, Doki generates two coordinated edits and executes them in sequence. The first creates the new definition, and the second replaces the original text with a reference to that definition:

```
{
    {
      "id": "edit-6a",
       "target" : {  "type" : "insert",  "selector" : {
            "after" : "paragraph_12" } },
        "newContent"  : "Character @Villain = a dark
           sorcerer with glowing red eyes",
       "description"  : "Created a new character definition
           from selected text"
    },
    {
      "id": "edit-6b",
       "target" : {  "type" : "paragraph",  "selector" : { "
           nodeId": "paragraph_12" } },
        "newContent"  : "The @Villain raises his staff to
           cast a spell",
       "description"  : "Replaced original text with
           reference to the new definition"
    }
}
```

9.2.3 Shot Generation Logic. algorithm 1 shows a pseudo code of the logic of deciding what shot and what media type to generate for each paragraph, depending on the shot states.

Algorithm 1: Shot Generation for Paragraph P

Input: Paragraph P with shots $s \in S$
Output: Generate button for P

if $\exists s \in S$ *currently generating (image or video)* **then**
  return Loading icon;
else if $\exists s \in S$ *can generate video* **then**
  return Clapper Board button (click to generate video for eligible shots);
else if $\exists s \in S$ *can generate image* **then**
  return Paint brush button (click to generate image for eligible shots);
else
  return no button;

Definitions:

A shot can generate video if it is not a definition shot, has an image ready (or outdated, or a saved image), and is not already generating.
A shot can generate image if its status is idle, ready, outdated, or error.

9.3 Supported Models

Doki by default uses Gemini 2.5 Flash for text, Flux Kontext Pro for images, and Veo 3 Fast for videos, while also supporting a broad range of generative models. For text-based interactions (image and video rewriting, AI agents), Doki supports Gemini 2.5 Flash and Pro. Image generation models include Imagen 4.0 Fast and Ultra, Ideogram 3.0, Flux Kontext Pro and Max, and Gemini 2.5 Flash Preview (nano banana). Among these, Flux Kontext Pro, Flux Kontext Max, and Gemini 2.5 Flash Preview accept images as context, while the remaining models operate solely on text. Video generation is enabled through Veo 2.0, Veo 3.0, Veo 3.0 Fast, and Runway Gen4. Veo 3.0 and Veo 3.0 Fast can produce synchronized audio. Veo models generate 8-second clips and Runway Gen4 generates 5-second clips.

9.4 Prompts

9.4.1 Image Rewriter: This module transforms structured prompts into vivid, detailed descriptions suitable for image generation. It takes in structured prompt and document context, and rewrites the prompt into expressive natural language while preserving fidelity to the source definitions. The rewriter ensures narrative consistency based on document context and adapts the description across

diverse content types, including character appearance (*e.g.,* will use plain background), settings, visual style, and camera shots.

9.4.2 Video Rewriter: The video rewriter performs an analogous role for video generation. It produces detailed, coherent prompts that expand the structured input into cinematic sequences. Unlike the image rewriter, which focuses on visual descriptiveness, the video rewriter emphasizes motion and temporal flow. Prompts specify subjects, actions, camera positions, compositions, and relevant audio when explicitly mentioned. We also leverage cinematography principles to guide shot design such as camera motion and composition. The video rewriter also includes negative prompts to prevent undesired elements such as background music, speech, or text overlays if not specified by users.

Id	Age	Gender	Role	Video Types	Video Editor Use
1	28	M	UX Designer	Professional films	DaVinci Resolve, GenAI tools
2	32	M	Filmmaker	Social media, Professional films	Adobe Premiere Pro, Final Cut Pro
3	24	F	Product Designer	Social media, Personal story or vlogs	Adobe Premiere Pro, iMovie, CapCut
4	57	M	Software Engineer	Personal Story or vlogs, Music video	Adobe Premiere Pro
5	28	F	Animator	School projects, Work-related content, Professional films	Adobe Premiere Pro
6	37	F	Program Manager	Social media, Personal story or vlogs	CapCut
7	24	F	Animator	School projects, Professional films	Adobe Premiere Pro, iMovie
8	29	F	Product Designer	Social media, Personal story or vlogs, School projects, Work-related content	Adobe Premiere Pro, Final Cut Pro, iMovie, CapCut, GenAI tools
9	38	M	Graphic Designer	Social media, Personal story or vlogs	Adobe Premiere Pro, DaVinci Resolve, iMovie, CapCut, GenAI tools
10	27	F	Content Creator	Social media	Adobe Premiere Pro, iMovie, CapCut

Table 2: Participants information including demographics, roles, video practices, and generative AI tool usage.

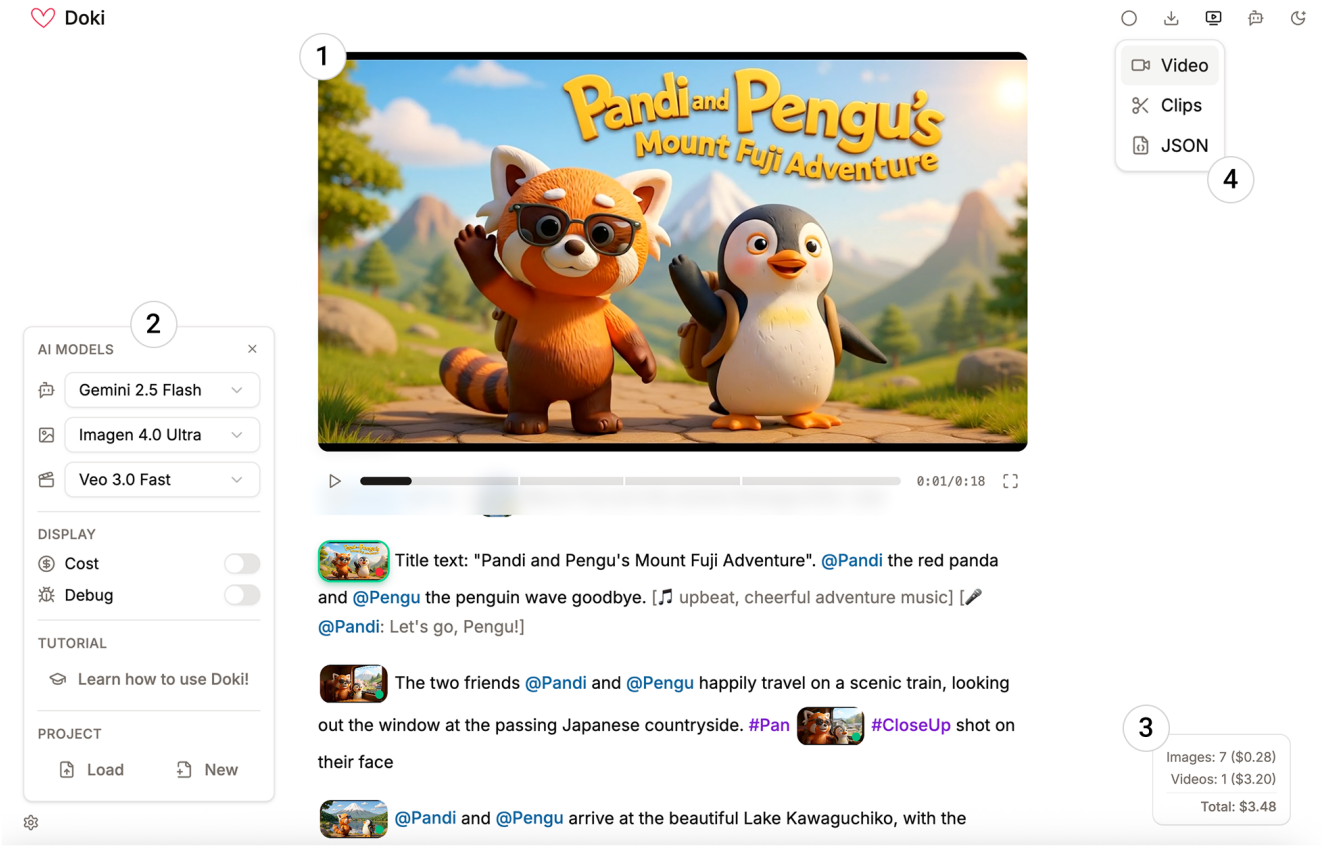

Figure 12: Additional features in Doki: (1) Preview video player shows the generated video with shot segments. (2) Settings panel lets users select AI models, toggle cost/debug options, access tutorials, and manage projects. (3) Cost monitor tracks generation usage and expenses across images and videos in real time. (4) Download menu provides output in video, clip, or JSON formats.

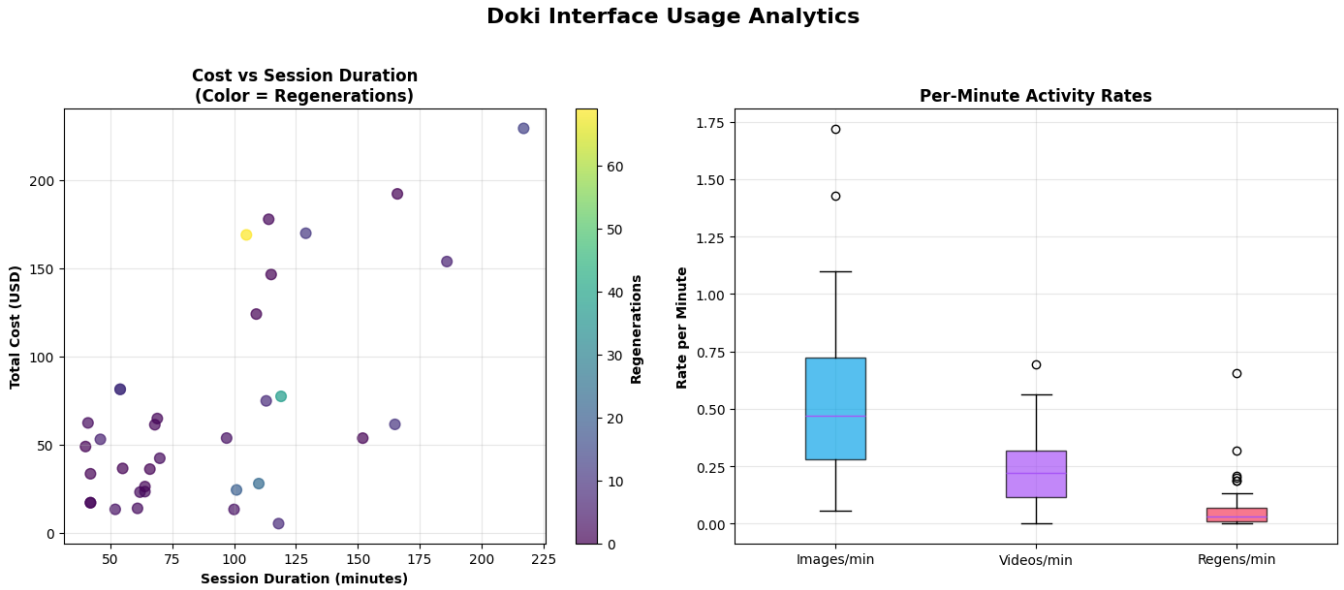

Figure 13: Doki interface usage analytics. Left: Total cost by session duration, with point color indicating regenerations. Cost data are available for 35 of 50 sessions, as some sessions were logged under ongoing projects rather than new ones as instructed. Right: Distribution of per-minute activity rates for images, videos, and regenerations. Images show the highest median rate, followed by videos, with regenerations least frequent.