

Notational Animating: An Interactive Approach to Creating and Editing Animation Keyframes

Xinyu Shi

University of Waterloo
Waterloo, Ontario, Canada
xinyu.shi@uwaterloo.ca

Li-Yi Wei

Adobe Research
San Jose, California, USA
lwei@adobe.com

Nanxuan Zhao

Adobe Research
San Jose, California, USA
nanxuanz@adobe.com

Jian Zhao

University of Waterloo
Waterloo, Ontario, Canada
jianzhao@uwaterloo.ca

Rubaiat Habib Kazi

Adobe Research
Seattle, Washington, USA
rhabib@adobe.com

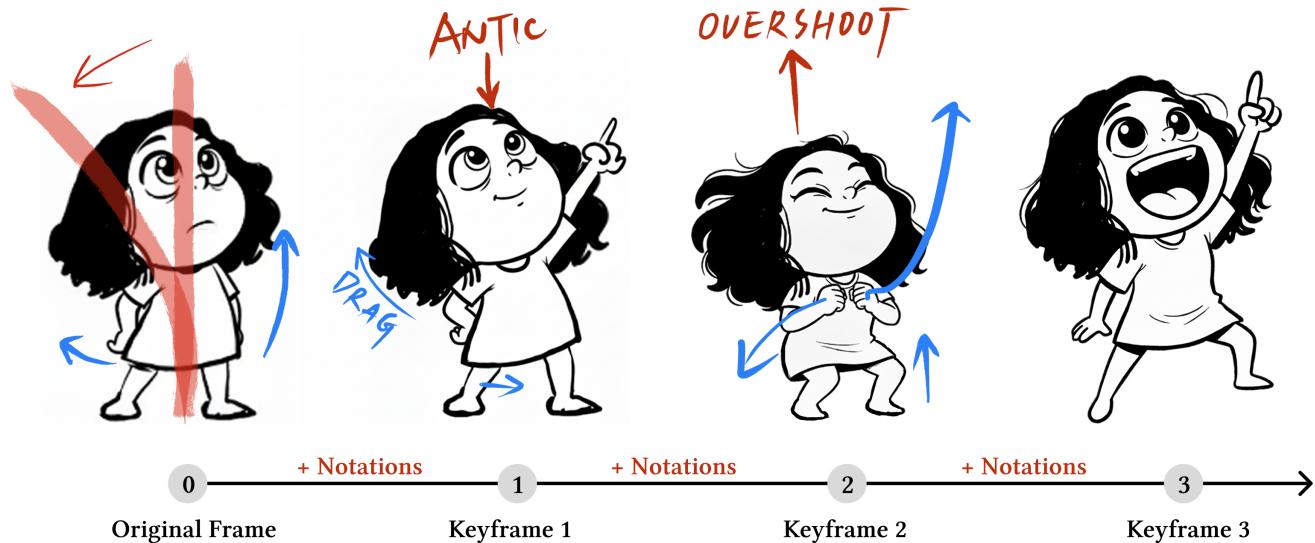


Figure 1: A usage example with *notational animating*: starting from a static drawing (left), the animator sketches high-level notations such as arrows, strokes, and text labels. The system interprets these notations and then generates the next keyframe. By iterating this loop (add notations and generate), the artist progressively authors and edits a sequence of keyframes (1–3).

Abstract

We introduce the concept of *notational animating*, an interaction paradigm for animation authoring where users sketch high-level notations over static drawings to indicate intended motions, which are then interpreted by automatic methods (e.g., GenAI models) to generate animation keyframes. Sketched notations have long served as cognitive instruments for animators, capturing forces, poses, dynamics, paths, and other animation features. However, such notations are often *contextual, ambiguous, and combinational*

based on our analysis of 135 real-world sketches. To facilitate interpretation, we first formalize these notations into a structured animation representation (i.e., source, path, and target). We then built an animation authoring system that translates high-level notations into the formalized intended animation, provides dynamic UI widgets for fine-grained parameter control, and establishes a closed feedback loop to resolve ambiguity. Finally, through a preliminary study with animators, we assess the usability of *notational animating*, reflect its affordance, and identify its contexts of use.



This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.

CHI '26, Barcelona, Spain

© 2026 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9723-8/26/04
https://doi.org/10.1145/3772318.3790707

CCS Concepts

- **Human-centered computing** → *Graphical user interfaces; Interactive systems and tools;*
- **Applied computing** → *Arts and humanities.*

Keywords

Animation, Sketch, Notations, Design Intent Expression

ACM Reference Format:

Xinyu Shi, Li-Yi Wei, Nanxuan Zhao, Jian Zhao, and Rubaiat Habib Kazi. 2026. Notational Animating: An Interactive Approach to Creating and Editing Animation Keyframes. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems (CHI '26), April 13–17, 2026, Barcelona, Spain*. ACM, New York, NY, USA, 27 pages. <https://doi.org/10.1145/3772318.3790707>

1 Introduction

Expressive animation, distinct from physics-based simulation and live action films, aims to depict imagined worlds that are not necessarily realistic yet believable and compelling [47]. The craft of expressive animation has evolved from classical flipbook-style hand drawings to computer-aided direct manipulations, and now towards generative AI-assisted authoring based on natural language inputs. However, the tacit knowledge of how to give a performance with movement, weight, timing, and emotion is long internalized and conveyed through visual language by artists [54]; natural language alone is too ambiguous to capture such nuanced intent.

Drawing on animators' traditional practice, we envision a future of animation authoring where sketched notations serve as the *reified user intention* [5, 37] to create and edit keyframes that can be subsequently used by automatic methods (e.g., recent frame interpolation models [16, 29, 36, 51]) to generate in-between frames, a concept we term as *notational animating*. Sketched notations have historically been rooted in animators' practice where arrows, lines, brief labels, and other informal notations are leveraged to externalize their thoughts, feel the dynamics in movements, and communicate with others [2, 18, 47, 52, 54]. These fast, intuitive, and expressive notations afford rich meanings such as forces, pose dynamics, paths, timing, and stylistic choices.

Sketch-based interfaces have been widely explored for animation support within and outside the HCI community for both animation authoring and modeling. In the HCI community, interactive tools such as Draco [26], Kitty [25], Energy Brush [56], Motion Doodle [48] map sketched notations to specific animation effects, facilitating the creation of kinetic textures, multi-subject causal illustrations, elemental dynamics, and character motions. However, these tools rely on *system-defined visual abstractions*, where the appearance and semantics of notations are strictly pre-determined. This enforces a fixed vocabulary that users must memorize, limiting the tools' generalizability to broader, improvisational animation scenarios. Computer graphics and vision communities explored modeling approaches [46, 60] to treat sketched strokes as *explicit* and *exact* constraints that deterministically drive trajectories or subject movements. However, in practice, animator's notations are often ambiguous and contextual: a line can signal arriving *near* a position, emphasize timing or weight, or invite a humorous detour rather than prescribe a pixel-level accurate path. Moreover, both lines of research usually handle one notation type at a time, overlooking how animators *combine* notations to specify multiple aspects of animation effects.

In *notational animating*, we define sketched notations as *user-defined visual abstractions* which are *contextual*, *ambiguous*, and *combinational* guides for animation rather than pre-defined, strict, and isolated constraints. By aligning more closely with artists' natural use of marks and symbols, we shrink the *gulf of envisioning* [44], the gap between a creator's mental image and its externalization.

However, the inherent ambiguity in notations shifts the burden to the system to interpret the intent, stretching the *gulf of execution* [21]; and it can also widen the *gulf of evaluation*, as users must understand how the system interpreted their notations to iteratively refine the output. Thus, beyond supporting user-defined notation, a *notational animating* system should also (i) provide a formalized interpretation layer to narrow the *gulf of execution*, and (ii) establish a tight feedback loop to narrow the *gulf of evaluation*.

As the initial step towards *notational animating*, we first analyzed 135 real-world animator-produced sketches, in consultation with two professional animators (with over 25 and 15 years industry experience in animation respectively), and we categorized the meanings encoded in animators' notational marks. From these observations, we propose a structured animation representation that formalizes the often informal notations for systematic interpretation. We then built a system probe with a two-level feedback mechanism to communicate the interpretation to users for quick confirmation and correction. Additionally, the system also exposes the lower-level parameterized control for motion range and timing through dynamically generated UI widgets to complement the high-level notations. To test *notational animating*, we conducted a preliminary user study with 7 professional animators to evaluate its usability, understand its affordance, and examine how it shapes animators' thinking and design workflows. Our findings reveal that *notational animating* enables a more holistic approach to keyframe authoring, shifting mental models from *doing one thing at a time* towards *thinking and composing multiple elements together*. We also identified practical constraints, most notably AI latency and performance, that raised model training considerations for the broader AI community. Finally, we offer design implications for exploring alternative animation representations across different abstraction levels, informing future research on AI-supported animation authoring.

In summary, we articulate the vision of *notational animating* in this paper, along with the following contributions as the initial step towards realizing this vision:

- The characterization of animators' notational practices;
- A formal animation intention representation maps the informal notations into system-executable actions;
- A *notational animating* prototype featured a two-level feedback mechanism and dynamic UI widgets for fine-grained control;
- The insights from a preliminary expert study detailing how professionals perceive and use *notational animating*.

2 Related Work

2.1 Interactive Tools for Animation Authoring

2.1.1 Sketch-based Tools. In HCI literature, there is a rich history of easy-to-use animation tools using sketching and direct manipulation that aims to make animation accessible to broader audiences. However, these tools predominantly rely on a **system-defined** approach: they pre-determine the mapping between visual abstractions and their semantics, requiring users to learn precisely *what* they can sketch and *how* to sketch within a rigid and procedural workflow which often complemented with UI widgets for parameter specifications. Prior work has primarily explored two types of abstractions: those addressing *space-time interactions* and those defining specific *animation effects*.

Space-time abstractions. One line of work addresses space-time interactions by mapping temporal relationships onto trajectory-based abstractions. K-Sketch [14] allows users to sketch object paths but relies on disparate UI widgets to specify transformations. Similarly, DirectPaint [38] adopts trajectory sketching for video annotation, using the path to propagate temporal changes. Some research [20, 48] extends this space-time abstraction into 3D character animations. These systems operate on a rigid keyframe-based logic where the stroke serves a single, fixed function (defining the path), while other properties, such as shape deformation, are managed via explicit parametric controls or algorithmic optimizations.

Animation effect abstractions. Other systems represent *specific* animation effects as distinct visual or gestural abstractions. Early work by Igarashi et al. [22] mapped multi-touch gestures directly to shape deformation. Later, Motion Amplifiers [27] reified animation principles (e.g., stretch and squash) as individual first-class visual objects which can be applied to drawings for exaggerated animation effects; and Energy Brush [56] utilized sketched arrow as the abstraction to depict secondary dynamics like fluid flow of smoke, fire, water and other particles. Such rigid mapping is also evident in Draco [26] and Kitty [25]. Draco mapped kinetic texture effects to strokes, requiring users to explicitly classify their input stroke as either *emitting* or *oscillating*. This taxonomy confines the output to the system’s predefined execution logic for translating strokes into animation. Kitty abstracts these entities into a graph view, enabling users to further specify the functional relationships through sketching and direct manipulations. Particular to character postures, prior work [19] adopted the line abstraction to quickly block out the shape of postures [19]. The recent work Squidgets [28] is perhaps the closest to our work, expanding stroke abstractions to support versatile effects like shape deformation, character rigging, and motion poses. However, this expansion introduces significant ambiguity. To manage this, Squidgets requires *explicit* mode-switching: users must *pre-select* a category (e.g., abstraction curves vs. rig curves) before drawing. This approach constrains users to the system’s functional affordances and inhibits the combination of effects. Furthermore, ambiguity still persists even within categories (e.g., a shape stroke could imply scaling, moving, or deforming), yet the system offers only the *best guess* without a feedback mechanism to resolve it.

While we draw inspiration from these prior works, they remain limited by **system-defined abstractions** that tailored to *specific* animation effects with *pre-determined* meanings that users must memorize. In contrast, we aim to provide an expressive **user-defined abstraction system** that leverages generative technologies to *infer* meaning from *generalizable* animation contexts, rather than enforcing rigid mappings, and to design a transparent feedback mechanism to resolve ambiguity dynamically.

2.1.2 Generative Tools. The emergence of generative models has led to new approaches in animation. Consumer tools such as RunwayML, Google Veo, and Adobe Firefly are capable of generating videos and animations from static images (functioning as keyframes) and/or video prompts. Although these methods are powerful and can produce high-quality results, they offer limited fine-grained control for stylized and expressive animations.

To facilitate control, conditional motion synthesis techniques [55, 60] generate detailed 3D character animations from 2D sketches. Alternatively, some research [43] offers limited control by providing a library of pre-made animation templates, which users can select to automatically retarget motion onto their static drawings. Generative video in-betweening frameworks [16, 36, 46, 51] offer versatile controls that encompass keyframes, trajectories, and masks to guide the generated frames. However, their outputs are pixel-level representations, which present challenges for editing and iteration. Moreover, the input trajectories function as rigid constraints, in contrast to our proposed notational animation inputs that accommodate greater abstraction, flexibility, and ambiguity. Although these methods show potential for creating detailed in-between frames, our work prioritizes the generation and manipulation of keyframes, which then serve as the foundation for these techniques.

In the field of HCI, researchers have explored the application of Large Language Models (LLMs) for generation and verification [31] of animations from user prompts. This is achieved by generating animation code, rather than pixels, which allows for subsequent manipulation through code editing [49], keyframe editing [59], or dynamically synthesized UI widgets [30]. More recently, Narrative Motion Blocks [6] demonstrates how direct manipulation input can be augmented with textual prompts to effectively create and edit animations. These recent works have shown the promising opportunity offered by generative models to shift from low-level parameter tuning to high-level intent specification. Yet, text is often ill-suited for capturing precise animation nuances. Although hybrid approaches like Narrative Motion Blocks [6] combine text with direct manipulation, it is still an open question to determine the most effective interaction modality to communicate with generative models. We argue that sketch-based input, with its well-established benefits in HCI, offers an intuitive and effective communication channel. However, a critical barrier, lies in shifting from the rigid, system-defined mappings of prior tools to a paradigm of *user-defined abstraction* that supports *generic* animation rather than specific effects. Our work addresses this gap by introducing a notational animation framework where users express intent primarily through sketched notations, supplemented by direct manipulation when precise parametric control is required.

2.2 Expressing Intent beyond Natural Language

Natural language alone is often insufficient to effectively convey intentions to AI [7, 42, 45, 53]; Subramonyam et al. terms this the *gulf of envisioning* [44]. This reflects an *instruction gap*: generative models are highly sensitive to phrasing, while people routinely use varied expressions to convey the same meaning.

A large body of recent work replaces or complements textual prompts with sketches, marks, and gestures to clarify intention. TaleBrush [9] uses sketch lines to indicate narrative transitions, PromptPaint [8] provides a paint-like interface for semantic prompt interpolation, and Block-and-Detail [39] guides image generation with sparse strokes. In animation production, Kaur et al. [24] introduces a technique to overdraw on shots to communicate detailed hair and cloth motion for consistency across artists. Gesture-based systems express edit intent with masks for inpainting [4], colored

strokes for recoloring [58], and point dragging to adjust pose or facial expressions [35]. In programming, sketch-based interfaces have proven promising: Code Shaping [57] supports the sketch-based expression of program intent, while Notational Programming [3] bridges handwritten and typewritten notations.

Another growing line of work augments natural language with direct manipulation by reifying user intent as manipulable GUI elements: DirectGPT [32] lets users drag visual elements into prompts, DynaVis [50] integrates NLIs with dynamically generated widgets for visualization authoring. Design tools like Brickify [40] and AI Instruments [37] reify intent into manipulable tokens or instruments for image composition and editing, extending Instrumental Interaction to AI contexts [5]. Piet [41] enables direct manipulation for color authoring in motion graphic videos.

In this work, we aim to support intent expression in animation authoring by using high-level sketched notations as the primary modality, complemented by dynamically generated UI widgets to balance simplicity and fine-grained control. Unlike prior work, we introduce an intermediate intent layer that formalizes user intent without constraining how notations are used.

3 Background and Design Study

In *notational animating*, we aim to leverage sketched notations as *user-defined abstractions*, empowering users to employ their own visual vocabulary for animation keyframing without *system-defined* constraints. In this section, we first provide necessary background knowledge and define the scope of this work. Then, we detail our methodology, covering both the content analysis and the expert interviews. After that, we present the key observations and findings from the design study. Finally, we distill the insights that inform our design of *notational animating*.

3.1 Scope and Terminology

Although sketched notations appear in both storyboards and comics which have been extensively studied [1, 11–13, 15, 33, 34], animation keyframing is fundamentally different from storyboarding and making comics. Following the definition in established animation book [54], we clarify the scope of our work by distinguishing animation *keyframing* from *storyboarding and comics*, and then explaining the terminology *keyframes* and *in-betweens* in animation practices.

Keyframing vs. Storyboarding. Storyboarding belongs to the *planning* stage of animation. It operates at the scene or semantic level, defining narrative flow, staging, and camera choices. Each storyboard panel represents a discrete story beat (e.g., “*the character enters the room*”). The space between panels implies *semantic* jumps that must be mentally filled by the viewer, for example, changes in action, position, or perspective. Keyframing, by contrast, belongs to the *production* stage and operates at the motion or parameter level. A keyframe specifies the movement property of an object (e.g., position, rotation, pose) at a precise time. The gap between keyframes is not a narrative leap but a continuous, computable interpolation of movement. Keyframing begins only *after* the storyboard has established the story and major poses.

Keyframing vs. Making Comics. Comics are a distinct visual medium that aligns more closely with storyboards than with animation. Comic artists primarily convey *what happens* (i.e., narrative

beats), rather than *how motion unfolds*. For example, depicting a character running in a comic requires only a single image; the full cycle of weight shifts and stride variations is unnecessary. In animation, however, such nuances are essential: the cadence of a run is important to shape the character’s personality such as the iconic gait of *Donald Duck*. These fine-grained motion details in keyframing are typically omitted in comics.

In summary, storyboards and comics share many similarities and aim to establish narrative structure, whereas keyframing focuses on designing the precise mechanics and style of motion. Consequently, prior studies [1, 11–13, 15, 33, 34] on notations in storyboards and comics cannot be directly generalized to keyframing practices, which we will discuss in Section 3.3.

Keyframes vs. In-betweens. Grounded in professional practice [54], *keyframes* refers to key poses that make an animation believable, which could be classified into three types: (i) *keys*: the essential storytelling poses required for the action to read; (ii) *extremes*: the turning points (e.g., contacts, anticipations) that define the limits of the movement; and (iii) *breakdowns*: intermediate poses that determine *how* the action travels (e.g., arcs, spacing) to ensure believability. The typically workflow is iterative: animators typically draw *keys* first, then iteratively add *extremes* to structure the transition and *breakdowns* to shape the motion’s feel. In this work, we collectively refer to all three categories (i.e., keys, extremes, breakdowns) as *keyframes*, which require artistic decisions from animators. In contrast, *in-betweens* refer to all remaining transitional frames to make the motion fluid, which can be automated by computational interpolation methods. In *notational animating*, users draw the *keys* as base drawings and iteratively add notations onto them to generate *extremes* and *breakdowns*. The resulting set of *keyframes* is then interpolated by computational methods (e.g., a video model) to synthesize the *in-betweens* for the final animation.

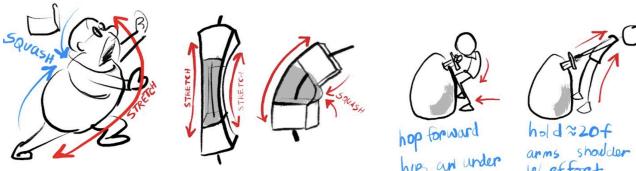
3.2 Design Study Methodology

Our design is informed by domain insights obtained from three sources: (i) established practices illustrated in the influential book in animation *The Animator’s Survival Kit* [54]; (ii) a content analysis of animation notations across 135 real-world sketches; and (iii) interviews with two professional animators with 25 and 15 years of industry experience respectively (noted as E1 and E2) to validate and contextualize our observations.

3.2.1 Data Collection. We collected a corpus of notation-embedded animation sketches (135 in total) from the online platform *Pinterest* where animators share work and foundational books [17, 54]. We searched using the key terms including “*animation sketch*”, “*motion sketch*”, “*animation notation*”, “*animation notes*”, “*pose notation*”, “*animation action sketching*”. We selected the sketches that contain at least one graphical notation (e.g., arrows, lines) and excluded examples containing only textual descriptions. To capture diversity across contexts, we balanced entities such as human characters, animals, cartoon characters, objects, and particle systems (i.e., water, fire, cloud, etc.). To reduce redundancy, we removed near-duplicates; for instance, among many sketches depicting walking in similar notating ways, we only keep the most representative one. The resultant corpus of the 135 examples can be found in the Appendix C.

Geometric Guides (*Hidden from Audience*) (Our Focus)

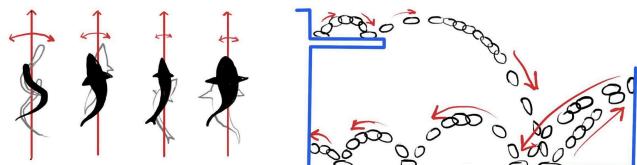
Force often notated with arrows, showing the cause of motion: the direction of the force, the position of the arrow's head or tail (sometimes could also be the proximity of the whole arrow) often indicate where the force should apply to, and sometimes relative magnitude. They also cue effects such as squash–stretch, drag, and follow-through.



Pose is also known as *line of action*. A single S- or C-curve overlays the figure to abstractly summarize a pose's dynamism and shape, coordinating multiple body parts without depicting each individual part separately.



Path could be notated with curves, dotted trails, and arrows, indicating trajectory and/or direction. These marks can be combined; for example, a main path with small arrows for secondary motion. In animation, the goal isn't merely to move an object along a line, but to make the motion feel believable or deliberately exaggerated, often via secondary motion (e.g., a whale's tail) and deformation (e.g., a bouncing ball). Hence, we categorize it under geometric guides rather than spatial guides.

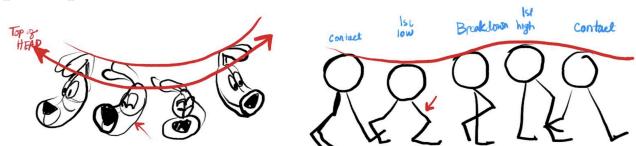


Style refers to the styling of the notations themselves (e.g., sharp v.s. soft curves, jitter, thickness) signals the intended feel of the motion.

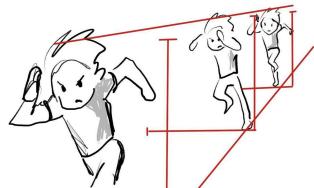


Spatial Guides (*Hidden from Audience*)

Arches (proportion and alignment) are curved lines traced across frames to keep parts moving on a single arc path and in correct proportion. They help to produce smooth and natural motion by aligning key poses along the same arc (e.g., head–ear–face during a turn; an arm swing while running). This is also known as the *arc* principle in animation.

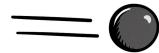


Perspective guides (depth and scale) include “rails” or straight converging lines tied to a ground plane or vanishing point. They show how position and size should change over time as objects move in depth or the camera shifts, preserving consistent spacing, foreshortening, and volume.



Motion Amplifier (*Visible to Audience*)

Motion lines are straight or arc strokes trailing behind a mover that indicate its path and suggest speed.



Impact stars are starburst marks at the contact point to show the instant of collision.



Circumfixing lines are short ticks ringing a figure or body part to show vibration, wobble, or small repeated motion.



Suppletion lines are parts of the figure are replaced by streaks, creating a blur that implies rapid motion.



Backfixing lines are lines fixed to the background (not the mover), creating a background blur that implies the camera is following the subject.



Partial polymorphism is only a body part is repeated to show its movement through time.



Figure 2: Three roles for animation notations emerged from the content analysis: geometric guides, spatial guides, and motion amplifiers. All visuals are redrawn by the authors to preserve IP while faithfully reflecting the originals.

Due to copyright constraints, we provide external links for all examples rather than displaying the images directly.

3.2.2 Coding. Prior to coding, we reviewed the book [54] telling the tacit knowledge in animation to ground our domain understanding and the nuances animators consider. Informed by this, the first author conducted inductive open coding of each example, annotating all notations and their intended meanings. Then, we employed an iterative procedure until no new codes emerged. After each round, the co-authors reviewed code meanings and groupings together; the codebook was refined through discussion. Following the iterations, the first author applied the codebook in a deductive recoding of the full corpus to ensure consistency. After that, we interviewed two professional animators (E1, E2) to examine the corpus, code definitions, and groupings, suggest clarifications, and confirm that no obvious notation types were missing. We incorporated their feedback and completed a final re-coding pass. We include the detailed codebook in Appendix B and the full coding results in Appendix C. The outcome of this analysis defines a broad framework and categories of animation notations.

3.3 Observed Notation Types and Meanings

Within the defined scope of animation keyframing process, we now consider the notational marks that often leveraged in keyframe drawings. In practice, animators augment drawings with quick annotations to externalize ideas and communicate with collaborators. As E2 noted, “*I love to use it to quickly capture my thoughts ... this way is intuitive and fast, letting me plan the whole and single decisions across parts.*” However, these notations are diverse and context-dependent. E1 emphasized that “*this kind of language (notation) varies across studios; conventions could differ.*”

To examine these practices, we conducted a content analysis of 135 animator-produced keyframe sketches (Section 3.2) and distilled recurring patterns in how motion and intent are marked. In specific, we categorized them into three groups: *geometric guides*, *spatial guides*, and *motion amplifiers*. It should be noted that we do not intend to propose formal definitions or definitive categories here. Our aim is to depict the scope and highlight identified recurring patterns (Figure 2) to support a shared understanding for animation notations. We describe the three identified categories below; detailed descriptions of each sub-category are embedded in Figure 2 alongside representative examples from our corpus. To avoid intellectual property concerns, all visuals were redrawn by the authors, with notations that faithfully reproduce the originals. The full corpus with the coding results can be found in Appendix C.

3.3.1 Geometric Guides. Geometric guides are notations that specify how an object’s shape should deform or a subject’s kinematics should change to realize an animation. This is the most common type of notation we found in our corpus (with 102 instances). We categorize these guides into four main purposes: force, path, pose, and style (the most representative examples for each category are shown in Figure 2). For each category, we further code them into detailed meanings, the detailed coding scheme can be found in Appendix B. These notations are planning aids for animators and will be removed before the final render; audiences never see them. We summarized the following key characteristics (noted as C1-4).

C1: The semantics of notation are highly contextual. Across the four categories (i.e., force, path, pose, and style), animators most often use arrows, lines, and curves, and occasionally other shapes (e.g., circles, rectangles, triangles). Yet these marks have no fixed *form – meaning* mapping: the same mark (e.g., an arrow) can represent different concepts across categories. Text, numbers, or color are sometimes added to clarify motion or timing, but these cues are often omitted and interpreted contextually (e.g., numbers may indicate sequence or a region of interest). In summary, unlike linguistic systems constructed by rigid syntax and grammar, animators’ notations are flexible and derive meaning from context.

C2: The usage of notations is often combinational. First, multiple notations conveying the *same* category of meaning (e.g., all related to force) can appear in a single drawing to depict individual motions for different parts. For example, Figure 3.a shows how the body stretches and squashes at the same time. Second, notations representing *different* categories of meaning (e.g., force vs. pose) can be combined to depict multiple aspects of an animation such as Figure 3.b. Third, multiple notations can be used in combination to serve *hierarchical* roles, such as the primary motion in combined with secondary motions, as shown in Figure 3.c.

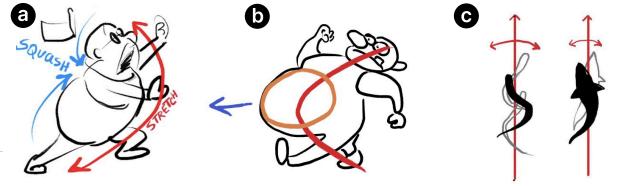


Figure 3: Combinational usage of notations: (a) combining notations all related to *force*; (b) combining notations of *pose* and *force*; (c) combining notations for *primary motion* and *secondary motion*.

C3: The meanings of notations are inherently ambiguous. This ambiguity manifests in three primary aspects. First, regarding **scope**, notations are often placed near a specific body part to indicate where the animation applies without explicitly specifying the target object. Thus, the affected area can only be inferred from context: it could refer to the entire character, a local limb, or even multiple connected parts. Second, regarding **spatial precision**, notations are expressive in describing high-level intent about what the motion is and how it feels, but rarely specify *pixel-level* accuracy. For example, an upward arrow indicating a jump with its rough intensity but does not necessarily mean the exact height which instead “*could be iteratively tuned to find the right one (magnitude).*” -E2 Lastly, ambiguity arises in **timing**. Numerical annotations can indicate action order, but animation timing is more complex, requiring coordination of durations and overlaps. Such temporal ambiguity likely because static spatial marks are short in conveying duration or speed. As noted by E1, these details remain abstract: “*they are acting in my head but I won’t write it down.*”

3.3.2 Spatial Guides. Unlike geometric guides that mostly annotate a single frame, spatial guides often span multiple frames. They map how positions and scales should change over time and align

arcs, paths, and sizes across frames with lines or curves. Like geometric guides, they are temporary construction aids and are removed before the final animation. There are two primary categories in spatial guides: proportion alignments and perspective guides. We explain them in details with representative examples in Figure 2.

3.3.3 Motion Amplifiers. As this category overlaps significantly with prior studies in comics [1, 11–13, 15, 33, 34], we adopt their established taxonomy and show the examples for the six different kinds of amplifiers in Figure 2 but omit a detailed review here. The key difference distinguishing motion amplifiers from geometric and spatial guides is their function: amplifiers help the *audience* feel the motion effect in a static medium, rather than assisting the *creator* in thinking about the motion. Thus, notations for geometric and spatial guides are never visible to the audience, whereas motion amplifiers are deliberately created for audience.

3.3.4 Focusing on Geometric Guides. In this paper, we narrow down our focus to support *geometric guides*. We exclude *motion amplifiers* because of two reasons. First, they function primarily as *audience-facing* perceptual cues, helping viewers perceive motion in static media like comics, rather than as *creator-facing* tools for thinking and planning motion dynamics. This distinction was confirmed by experts E1 and E2. Second, even if *motion amplifiers* could be re-purposed for authoring, integrating them will result in a *system-defined notational system* where users have to learn and select from a fixed library of symbols. It is because they are well-established visual conventions in comics with deterministic meanings for each. This conflicts with our goal in this work to enable *user-defined notations* that users can flexibly define their own notations in free-form. We thus limit our focus to notations that stay informal and externalize animator's intent directly.

Additionally, while *geometric* and *spatial guides* assist in motion planning, they serve distinct roles: *geometric guides* operate *locally* to define the internal structure of individual keyframes, whereas *spatial guides* operate *globally* to define composition across multiple keyframes. We argue these processes are functionally decoupled and present different interaction needs. Spatial alignment can be intuitively handled via direct manipulation (e.g., move, scale, and rotate existing keyframes). In contrast, crafting the geometric deformations of animated subjects, such as designing their poses, requires much creative efforts. While AI could theoretically assist in spatial alignment as well, the current bottleneck for animators remains in the *generation* of these individual frames, not their *placements*. Therefore, to avoid over-complicating the problem space, we prioritize the most critical problem of generating individual keyframes that align with the animator's intent expressed through *geometric guides*. We reserve the integration of *spatial guides* for future iterations, with an envisioned direct manipulation-based interaction approach similar to StickyLines [10].

3.4 Design Goals

Based on above observations from animators' sketches, with a focus on the *geometric guides*, and our discussion with professional animators, we distilled the following design goals to guide the design of a *notational animating* system.

DG1: Support freeform notation without rigid syntax. To accommodate the freeform nature of sketching, the system should allow users to notate flexibly (*C1: contextual semantics*) without enforcing a fixed vocabulary or grammar. That is, how to notate should be decided by user themselves instead of the system.

DG2: Accommodate combinational usage for multifaceted intentions. As animation design decisions are inherently intertwined, users often express multiple intentions within a single drawing. The system must support arbitrary combinations of notations (*C2: combination usage*), enabling it to recognize and process multiple constraints (e.g., force and pose) simultaneously.

DG3: Provide structured interpretation and transparent feedback. Given the contextual dependency and inherent ambiguity of notations, using them to instruct an automated system requires transparency. Thus, the system must establish a closed feedback loop that communicates the AI's interpretation to users, including *what* the notations mean (*C2: contextual semantics*) and *how* they apply to the animated subjects (*C3: ambiguous scope*). The feedback should minimize cognitive load for at-a-glance verification, with intuitive mechanisms for quick error correction.

DG4: Facilitate iterative tuning through complementary interaction modalities. Since notations fall short in conveying specific spatial magnitudes and temporal durations (*C3: ambiguous spatial/temporal precision*), the system should provide interaction modalities beyond notation sketching to allow users to iteratively adjust motion range and timing.

4 Notational Animating

Informed by the characteristics of animators' notations (*C1-3*) identified in our design study and the distilled design goals (*DG1-4*), we define the concept of *notational animating* and articulate its principles. We then introduce a *formalized animation representation* for systematically interpreting notations and describe key features in a *notational animating* system.

4.1 Definition

We propose *notational animating*, an interaction paradigm for animation keyframing, in which users sketch *high-level notations* over static drawings to indicate intended animations. With the reasoning capability, a Vision-Language Model (VLM) interprets these notations into structured intent, which is used by a generative model (e.g., an image generation model) to synthesize animation *keyframes*. The resulting *keyframes* can then drive computational interpolation approaches, such as recent video models, to produce *in-between* frames for the final animation. In the following, we introduce the **scope** of *notational animating*, the **role** of notations, and the **interaction modalities** in a *notational animating* system.

- Scope: a generalizable interaction paradigm.** We intend to design *notational animating* for broad and general animation authoring instead of specific animation effects, including character animation, particle system animation, logo animation, etc. Thus, when interpreting the notations, any domain-specific assumptions should not be made. For example, it should not assume a set of possible animations and select the most closed one for a notation, but all meanings should be inferred from the context.

- **Role: notations as user-defined abstractions.** The notions in a *notational animating* system, serving as *user-defined abstractions* to express intended animation, should stay *informal*, in contrast to the *formal notations* in math, physics, and music where a universal convention should be learned before using. The notations in a *notational animating* should also be *contextual, ambiguous, and combinational* as discussed in Section 3.
- **Interaction: sketching combined with direct manipulation.** In a sense, *notational animating* does not mean that users should only rely on sketching notations. We recognize that notations cannot afford all intentions. Thus, in a *notational animating* system, to fully realize artists' vision, notations should be used together with direct manipulation. Notations are for high-level expression and direct manipulation is for low-level control, especially on the motion spatial range, effect intensity, and timing coordination.

4.2 Structured Animation Representation

Instead of formalizing notations, we choose to formalize the intended animation representation to facilitate the interpretation of the notations (*DG1: support freeform notation*).

Animation Notation Representation:

Primary Notations: $\langle \text{Source}, \text{Path}, \text{Target} \rangle$ to express animation semantics:

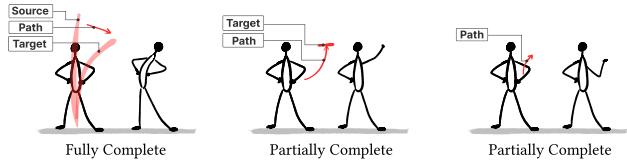
Source: where motion originates, what components involved to move, could be implicitly indicated by proximity or explicitly marked by users.

Path: how motion unfolds, it could be [force | trajectory | direction | etc.], while often shown by arrows, it can also be conveyed through lines without arrowheads.

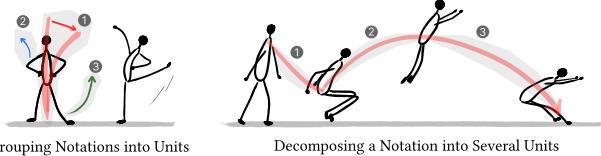
Target: how the intended end status looks like or where the motion ends. The line of action could serve as one kind of "target", but other forms are possible, such as a simple positional mark indicating where the source should move to, or the drawing of a detailed pose.

Secondary Notations: to express stylistic descriptors of an animation and add additional information or clarifying primary notations. [color | thickness | numeric labels | text labels | etc.] could be used to convey temporal order, motion style, intensity, or other expressive metadata.

This representation is based on observations from our corpus (**C1: contextual semantics**). We categorize notations into two types: primary notations and secondary notations, according to their appearing frequency and function. Primary notations consist of frequent graphical notations (e.g., arrows, lines, curves) that define the core *semantics* of the animation (e.g., force, pose, path). However, since notation semantics can be ambiguous regarding the concrete animation behavior and the targeted object (**C3: ambiguous scope**), we decompose the meaning of primary notations into a structured triplet: $\langle \text{source}, \text{path}, \text{target} \rangle$. Secondary notations consist of auxiliary but often omitted elements (e.g., color-coding, text, numbers) that provide *supplementary information*, such as temporal ordering or the expressive feeling of the motion. We highlight three key considerations during the interpretation process:



(a) Illustration of the three core fields in our formalized notation—source, path, and target. Left to right, examples vary in how completely these fields are specified. When any field is omitted, the system infers the missing information from context.



(b) Illustration of grouping (left) and decomposing (right) notations into multiple animation units.

Figure 4: Illustration of flexible completeness and the demonstration of multiple animation units in one sketch.

• **Visual Form Agnosticism:** This representation places *no constraints on the visual form* of notations (whether it is an arrow, a line, or a circle). Instead, it prioritizes each notation's functional role (i.e., source, path, target) in the intended motion (**C1: contextual semantics**). Additionally, this representation supports compositional notations: rather than assuming a one-to-one mapping between a single mark and an animation, it recognizes that a single animation unit is often composed of multiple distinct notations (**C2: combinational usage**). For example, a user may use a circle to specify the subject (source), an arrow to define the trajectory (path), and a line to indicate the final position (target).

• **Atomic Definition:** This representation *declares a single animation unit*, though users may specify multiple units simultaneously (**C2: combinational usage**), as shown in Figure 4b. Thus, during interpretation, notations should be first grouped into units, each unit is then interpreted with this representation (*DG2: accommodate multifaceted intentions*). On the other hand, a single notation may denote multiple units (e.g., in Figure 4b, the leap consists of three units). Thus, during interpretation, a notation may need to be decomposed into multiple animation units.

• **Flexible Specification Completeness:** This representation allows for *flexible completeness*. Users are not required to specify the full triplet $\langle \text{source}, \text{path}, \text{target} \rangle$, nor are secondary modifiers mandatory (**C3: ambiguous meaning**). When all elements are present, the system interprets the motion intent with high precision. Conversely, when elements are missing, the system accommodates *creative ambiguity*; the interpreter (e.g., vision-language models) infers the missing fields from the context.

This formalization serves two main purposes. First, it resolves ambiguity by requiring the system to explicitly infer essential components from the context. Our experiments indicated that without such structural constraint, VLMs tended to default to high-level, global semantic descriptions. In contrast, enforcing this structure guides the model to produce fine-grained interpretations that align

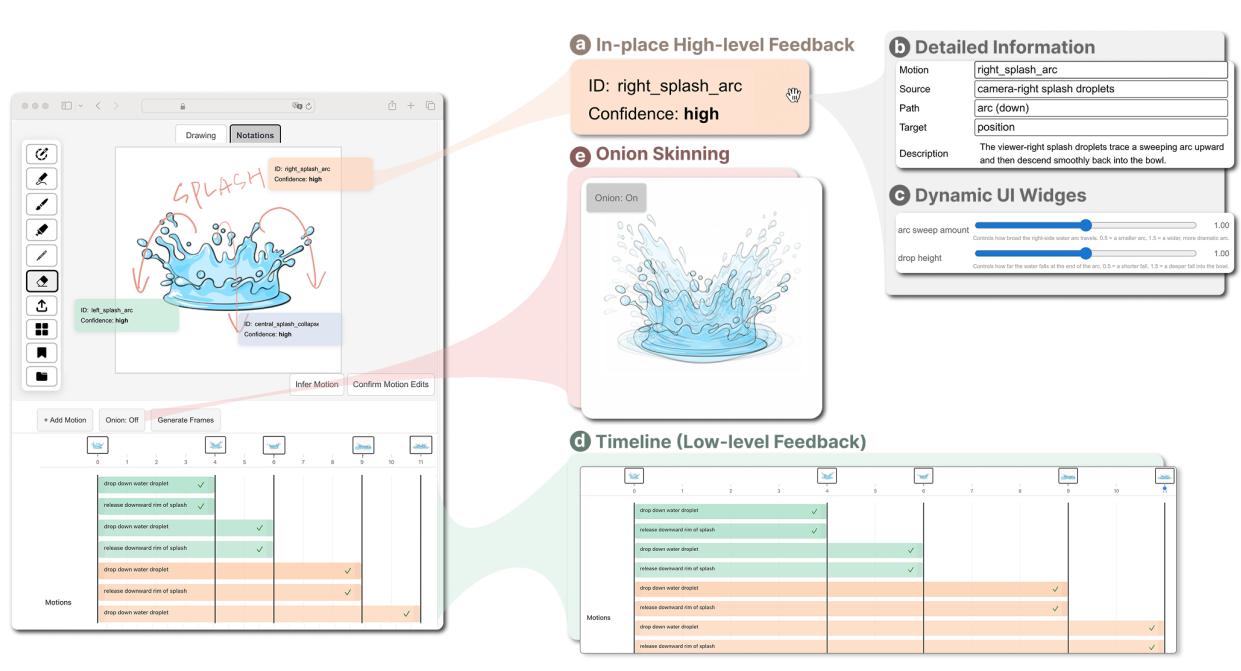


Figure 5: Interface of our *notational animating* system: the main canvas is augmented with (a) motion tags for in-place high-level feedback, (b) hoverable detailed information, (c) dynamically generated sliders for motion range adjustments, (d) an auto-constructed timeline for timing edits and per-object feedback, and (e) onion-skin overlays to review keyframe movement.

better with the animator's specific intent. We provide an example in Appendix D comparing VLM interpretations with and without this structural constraint. Second, such a formal structure could allow for providing users with the *structured and verifiable feedback*. Without an explicitly defined structure, interpreters return free-form text that varies unpredictably across different cases and models. This variability makes it difficult to visualize the AI's understanding effectively. Presenting users with lengthy, unstructured text imposes a significant cognitive burden, hindering their ability to quickly verify and correct the interpretation.

5 The Notational Animating Prototype

We implemented a prototype system as a testbed for *notational animating*. Our prototype consists of three main components: (1) a drawing canvas to sketch and iteratively refine notations; (2) a two-level feedback mechanism that communicates how notations are interpreted and resolve ambiguities (*DG3: provide structured feedback*); and (3) direct manipulation widgets (i.e., dynamic UI and timeline) for fine-grained control about how each individual subject parts move (*DG4: facilitate iterative tuning*).

5.1 Drawing Canvas

The canvas allows an animator to draw the static frame and note on top of it by switching between the *drawing* and *notations* modes at the top of the canvas. Besides drawing from scratch, users can also upload an image. The drawing and notating brush's size and color can be changed by the tools provided in the toolbar at the

left side of the canvas. The system treats the drawings and/or user-uploaded image in the same way as a rasterized image. The animator can save the current workspace (including all drawings, notations, status of the timeline, and results) by the *Save Frames* button in the toolbar and revisit previous ones in *History*.

5.2 In-place Motion Labels

After the user finishes drawing and adding notations, clicking *Infer Motion* sends the drawing or uploaded image together with notations to a vision–language model (VLM) for interpretation.

High-level feedback for animation interpretation (*DG3: structured feedback*). Each animation unit is returned as a colored *in-place motion label* (Figure 5a), with colors randomly assigned per unit, positioned adjacent to its corresponding notation(s). Each label shows a brief semantic description and a confidence rating (i.e., low, medium, or high) and can be dragged to reorganize the workspace. Hovering it will reveal details (Figure 5b) about its inferred \langle source, path, target \rangle and a brief description. We surface this information structure to give users an ambient mental model of how the system interprets their notations and to reason about how to communicate more effectively. These details are editable, enabling users to correct any AI misinterpretations. After making changes, the user clicks *Confirm Motion Edits*; the system incorporates the edits and re-infers the motion accordingly.

Dynamic UI widgets for fine-grained control. In parallel, the system generates context-aware slider widgets (Figure 5c) for fine-grained control over spatial range and motion intensity (*DG4: facilitate iterative tuning*).

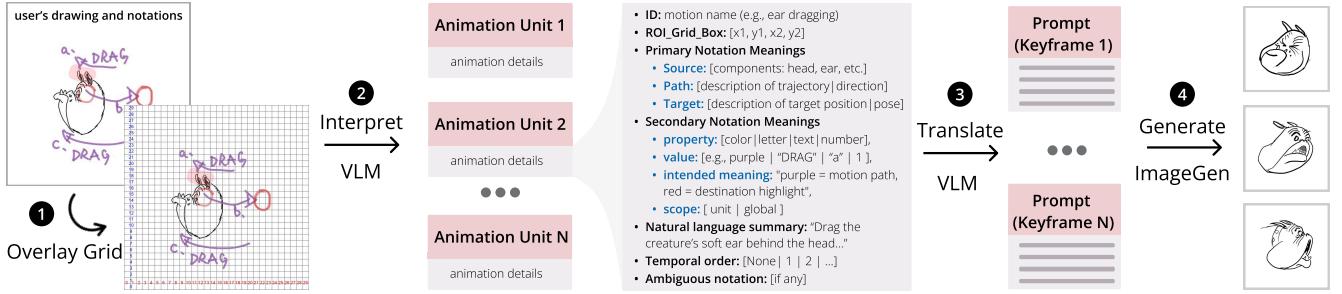


Figure 6: The technical pipeline of using VLM to interpret the sketched notations, consisting of four main steps: (1) overlay a grid onto user's drawing and notations; (2) VLM interpret it by identifying the animation units and interpret each unit in a structured way; (3) VLM translate the structured interpretation into natural language as the prompts for each keyframe; (4) parsing to an image generation model to generate keyframes.

iterative tuning). Default maximum and minimum values are provided. For quantitative parameters (e.g., rotation degrees), numeric values are shown; for qualitative ones (e.g., how high the hand raises), slider endpoints are labeled with semantic anchors (e.g., minimum = shoulder level). It is worth noting that alternative design choices exist for providing fine-grained control over motion intensity and spatial ranges. Ideally, the most precise and intuitive method would be *direct manipulation via on-canvas control handles*. However, integrating such interactions into current generative AI pipelines presents significant feasibility challenges. First, since the system operates on raster images, generating the necessary semantic handles such as kinematic skeletons for characters requires non-trivial engineering efforts. Second, even if users were to manipulate pixels directly, current generative models struggle to adhere to such precise pixel-level guidance without extensive fine-tuning. Thus, we adopt dynamic UI widgets as a pragmatic solution to offer fine-grained control within current technical constraints. We envision that future advancements will eventually enable more direct and precise manipulation approaches.

5.3 Timeline

The timeline (Figure 5d) is automatically constructed once in-place feedback is generated and serves twofold purposes.

Low-level feedback for how primitive parts move (DG3: structured feedback). For each *animation unit*, the system decomposes motion into per-part tracks. Each timeline block represents a primitive part (e.g., a character's head) and is labeled as *primitive's name - verb* (e.g., head tilt up). Timeline block colors mirror the in-place feedback labels for easy mapping: all per-part timeline blocks belonging to the same animation unit share the color of its in-place *motion label*. By hovering over each block, a detailed description about how this part moves will be shown.

Temporal order and duration adjustment. The timeline allows for direct manipulation of temporal order and duration (DG4: *iterative tuning*). Blocks can be moved along the time axis, resized to adjust duration by dragging their edges, deleted, or added. By default, a keyframe is appended to each block, shown as black vertical lines with thumbnail previews above the timeline; hovering enlarges a preview, and clicking places the keyframe on the canvas.

When satisfied with the interpretation and timeline, the user clicks *Generate Frames*; the system fills all empty keyframe placeholders with generated keyframes. By toggling on the *onion-skinning* (Figure 5e) button, user-selected keyframes will be overlaid on the canvas to visualize the animation.

5.4 Implementation Details

Our system utilizes a Vision-Language Model (VLM) to interpret user notations into structured animation intentions and an image generation model to generate keyframes. In specific, we use OpenAI's o3 API for (i) motion inference, (ii) decomposing each animation unit into primitive motions, and (iii) generating detailed keyframe descriptions. Keyframe images are produced with the gemini-2.5-flash-image-preview API. Our pipeline consists of four main stages (Figure 6).

Motion interpretation. To enable precise localization, we first employ a *spatial grounding* technique by overlaying a 30×30 coordinate grid on the input sketch. This grid provides a fixed reference system, enabling the VLM to ground observations in specific numerical coordinates rather than vague spatial descriptions. Next, the VLM decomposes the sketch into "animation units." We employ a structured prompting strategy (Appendix 10) to extract a formal representation (see Section 4.2) for each unit, returned as a JSON object containing unique identifiers and Region of Interest (ROI) coordinates for each unit. This structured output enables the interface to synthesize the interpretation tags with semantic data and utilizes the ROI coordinates to spatially anchor these tags adjacent to their corresponding notations.

Keyframe generation. Following interpretation, the VLM translates these structured specifications into natural language prompts optimized for image generation. By integrating inferred temporal logic with user-manipulated timeline constraints, the system synthesizes frame-specific prompts that detail both local component movements and global scene states. Finally, these prompts and the initial drawing will be sent to an image generation model. For multi-frame sequences, we adopt a progressive strategy: each generation step is conditioned on the previously generated keyframe and the target prompt of the next, iterating frame-by-frame to ensure temporal continuity and smooth motion.

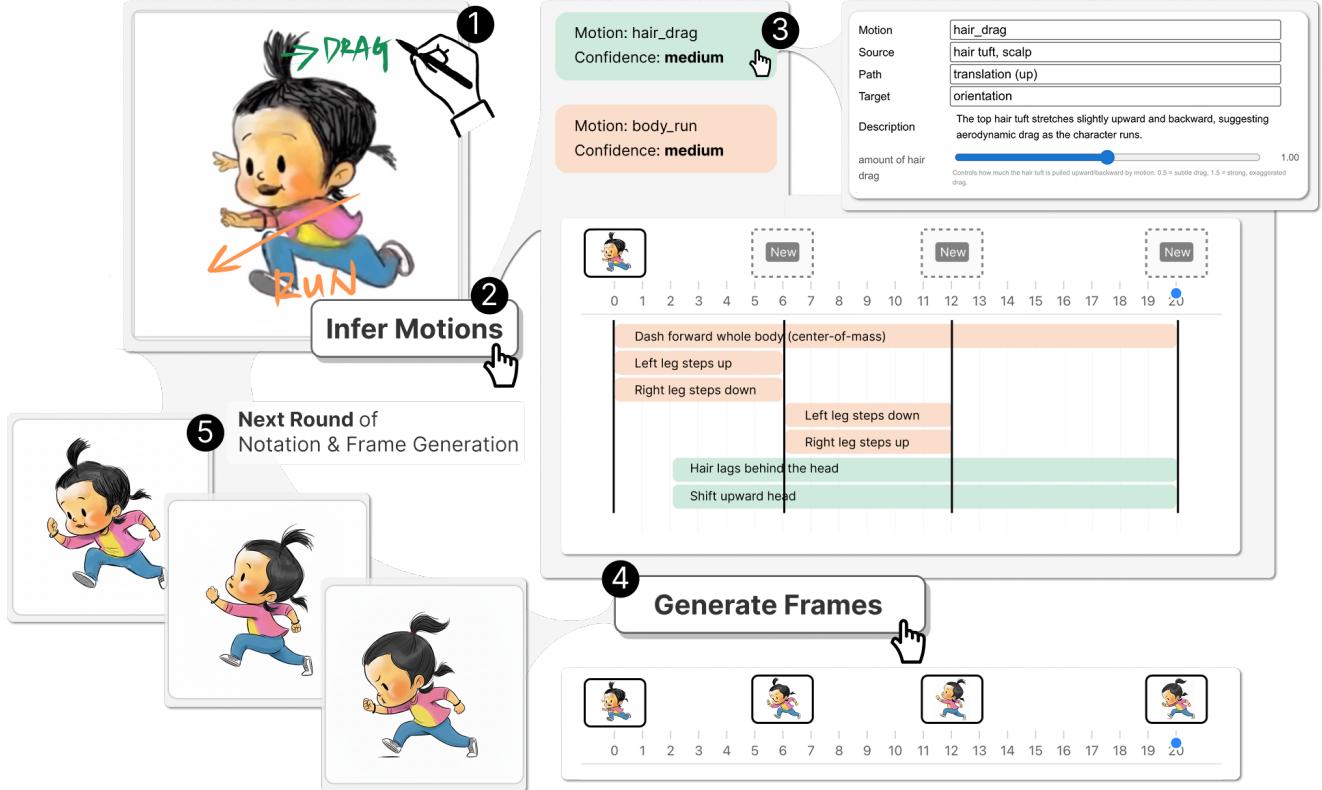


Figure 7: Workflow with *notational animating* for character motion: (1) sketch high-level notations on the drawing; (2) click “Infer Motion”; (3) the system overlays motion tags near the marks, hovering a tag reveals details, and automatically constructs a per-object timeline; (4) click “Generate Frames” to produce the next three keyframes; (5) continue notating on the generated frames to progressively author a longer keyframe sequence.

6 Demonstrative Examples

In this section, we present three illustrative examples produced with our *notational animating* prototype. We provide a detailed account of the creation process for the first example to showcase the *notational animating* workflow. For the second and third examples, we omit step-by-step details and instead highlight how the concept generalizes across diverse animation contexts and intentions.

Character Motion. This example illustrates a common consideration in animation design: a primary motion (running) accompanied by a secondary motion (hair drag), and demonstrates how to adjust the timing of their overlap. Starting from the initial frame, as shown in Figure 7-1, we place a long orange arrow labeled “RUN” to indicate the child’s primary motion and direction of the movement. We then add a green arrow near the child’s hair to denote the secondary motion: a slight delay that produces a follow-through drag. Next, we click Infer Motion (Figure 7-2). The system overlays two motion tags (Figure 7-3) near the notations: “body run” and “hair drag”; and hovering over a tag reveals more detailed information. The timeline is then constructed automatically, decomposing the two motions into seven tracks corresponding to individual primitive



Figure 8: Demonstration of using dynamic slider to adjust the motion range for the “DRAG” motion.

motions. The timing of the primary and secondary motion is often carefully overlapped, which we can specify by dragging and adjusting motion blocks on the timeline. In doing so, we specified the next three keyframes to generate. Clicking “Generate Frames” button (Figure 7-4) produces these three keyframes in sequence. Continuing to notate on the generated frames extends the sequence into a longer run of keyframes (Figure 7-5). Using the dynamic sliders, we can also adjust the range of the “drag” motion (Figure 8).

Table 1: The table records our interviewees’ demographic information, including genders, range of ages, occupations, experiences in animation creation in years, and self-rated expertise in animation design in 5-item Likert scale.

ID	Gender	Age	Occupation	Experience	Expertise (1-5)
P1	F	26	Creative Designer	4 years	4
P2	M	41	Sr Technical Research Artist	20+ years	5
P3	M	46	Sr Quality Engineer	10+ years	5
P4	M	32	Digital Artist	10+ years	5
P5	F	26	Indie Game Developer	8 years	3
P6	M	45	Designer	10+ years	5
P7	F	31	Ph.D. student	2 years	3

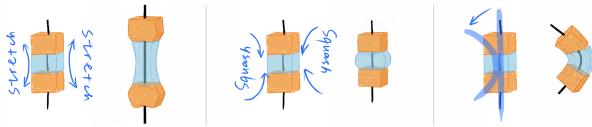


Figure 9: Object animation with *notational animating* illustrating squash-and-stretch and shape deformation.



Figure 10: Demonstration of creating animation effects for particle system such as water with *notational animating*.

Object Animation. This example (Figure 9) demonstrates how *notational animating* conveys dynamism in static objects through shape deformation. We animate a stack of three cubes with coupled motion to show how notations can coordinate movement across multiple objects. In the first two keyframes, arrows encode direction and squash-and-stretch. In the final keyframe, a broad overlaid stroke abstracts the transition from the initial to the target shape. In all cases, the VLM infers that cubes move together over time.

Particle System Dynamics. In this final example (Figure 10), we use the notation to drive a particle system animation of a water splash. The three panels illustrate a temporal progression from the moment of impact (left), through the peak of the crown splash (center), to its dissipation (right). Directional arrows specify the intended motion field for particles: outward arrows around the rim indicate radial flow along the water surface, while upward arrows convey the vertical momentum that generates the crown. Arrow length and curvature encode both the expected trajectory and relative magnitude, which together suggest the emergent shape of the splash. This example suggests the possibility beyond descriptive guidance, the notation can act as a lightweight control interface for physics-based effects. Interpreting arrows as velocity vectors allows a renderer to initialize emitter directions and speeds, while additional notations can parameterize features such as splash height, sheet thickness, and droplet spread.

7 Expert Evaluation

Using the developed prototype as a testbed, we conducted a user study with professional animators to evaluate the authoring experience of *notational animating*. We aim to answer the following research questions: 1) how professionals use and perceive *notational animating*; 2) how well a *notational animating* system can support their control over the animation keyframe editing and how users can handle the misinterpretations; and 3) how might *notational animating* impact user’s way of thinking.

7.1 Participants

We recruited 7 experts in animation creation (3 females, 4 males). The detailed demographic information is listed in Table 1. The studies were conducted remotely via Microsoft Teams. Participants accessed our system through a web browser using a pen-based input device: 2 were using iPad, 3 were using Movink, and 2 were using other tablets with the same functionality. Each participant received a \$50 USD Amazon gift card for the 90-minute session.

7.2 Task Design

We designed two complementary tasks to examine how professionals use *notational animating* and to identify limitations.

- **Constrained task (targeted effect):** Participants were given a starting static image and a brief natural-language description of a target animation effect, then asked to realize it using *notational animating*. Prompts included: (i) character: human celebrates a triumph; (ii) animal: frog jumps to mid-air; (iii) natural dynamics: water pours onto fire. We cover different animated subjects to examine how participants notate in different contexts.
- **Exploratory task (open-ended):** Participants sketched their own initial frame and freely explored any animation they wished. This task aims to uncover unexpected behaviors and limitations.

7.3 Procedure

We began by briefing participants on the study purpose, procedure, approximate duration, and data collection practices, then collected demographics and background on animation design. Next, we introduced the concept of *notational animating* and emphasized that participants were free to invent and use any notations. After a short walkthrough of *notational animating*, participants completed a hands-on warm-up with a predetermined image unrelated to the study tasks. Once comfortable, they performed the two tasks in sequence while thinking aloud. Finally, participants completed

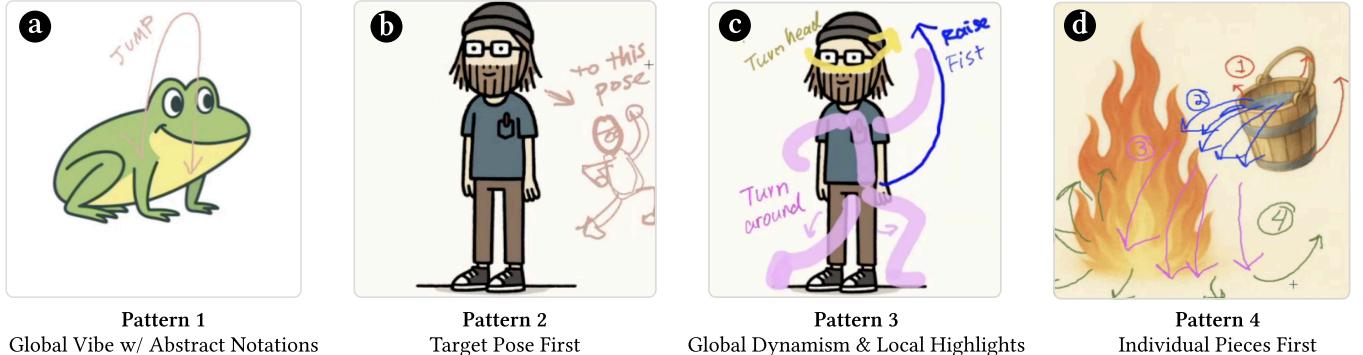


Figure 11: Representative examples of the four notating patterns identified in our content analysis. The patterns range from (1) expressing *global vibes* via *abstract notations*; (2) defining the *target pose* explicitly; (3) combining *global dynamism* with *specific local adjustments*; to (4) breaking down the scene into *individual pieces* or *sequential flows*.

a post-study questionnaire evaluating their experience with *notational animating*, followed by a semi-structured interview. All sessions were audio- and video-recorded. Participant-generated notational sketches, the transcribed think-aloud data and interview data were analyzed through inductive thematic analysis.

8 Findings

We report the findings in several themes: overall impressions of *notational animating*, notating patterns, failure cases encountered, how animators handle misinterpretation, how they make exaggerations and how their mental model shifts.

8.1 Overall Impression

Overall, our participants found this interaction concept is *impressive*, *useful*, and *intuitive*. On a 7-point Likert scale evaluating the statement “*It was easy to express my animation intent.*”, responses included: 2 strongly agree, 4 agree, and 1 neutral. Some of user-produced keyframes are shown in Figure 13. P6 reflected, “*I think this is really exciting and fun ... It was fluid, you know, animation is such a time consuming process and things like this get me excited about how much time they could save.*” Participants appreciated the intuitiveness of *notational animating* because they did not face a steep learning curve and felt their intention can be mostly captured, “*it was more like open to interpretation and like a quick and easy thing to do and that felt like I didn't have to learn any new language to do it. Such a way [notional animating] is approachable and exciting to me. I just draw arrows and things and write things on the screen and it seemed to interpret most.*” -P3

Participants emphasized that *notational animating* felt intuitive because it aligns with how they already think and communicate through sketching. P4 noted that, “*annotations are already part of how we work, when we revisit a piece or give someone direction, we mark things up to make the idea clear. It's how we visualize what to change.*” In contrast, they described prompt writing as a separate, learned skill that they had to learn “*its own language which is very different from ours.*” -P4 This perspective aligns with how artists are trained to develop motion progressively through sketching: “*The way we're taught is to start simple, block it out with basic lines and shapes, build the structure, then layer in detail,*

Table 2: Notating patterns adopted by each participant.

Participant	Patterns Adopted
P1	Pattern 1, 3, 4
P2	Pattern 1, 2
P3	Pattern 3, 4
P4	Pattern 2, 3, 4
P5	Pattern 1, 4
P6	Pattern 3, 4
P7	Pattern 4

and in this tool, I'm doing things in this flow.” -P7 Taken together, these comments suggest *notational animating* supports a natural interaction approach that builds on animators’ existing workflow.

8.2 Notating Patterns

8.2.1 Observed notating patterns across participants. We observed four recurring notating patterns from participants. These patterns span along two axes: *focus* (whole body or scene vs. individual parts) and *abstractness* (abstract intention vs. concrete specification). We regard notations (e.g., arrows) as more abstract while the user-drawn pose as more concrete. The same animator may switch patterns across tasks or even try different ways within a single task, we list how participants adopt different patterns in Table 2. Despite their different notating patterns, all animation intentions can ultimately be captured within our formalized representation. Below we describe each pattern with one representative example, the landscape of notation patterns can be found in Figure 12.

Pattern 1: Global vibe with abstract notations. This pattern reflects a process-oriented mental model where animators begin at a high level, focusing on *how the motion unfolds*. They typically sketch the overall motion arc or energy (via PATH) and add short semantic labels (e.g., *jump*). TARGET is often omitted and SOURCE is implicit (e.g., *the whole frog*). One notation often express multiple intended keyframes. For example, as shown in Figure 11.a, P6 use the arrow which consists of a small downward arc and a huge upward and downward arc to indicate “... (the frog) would be getting

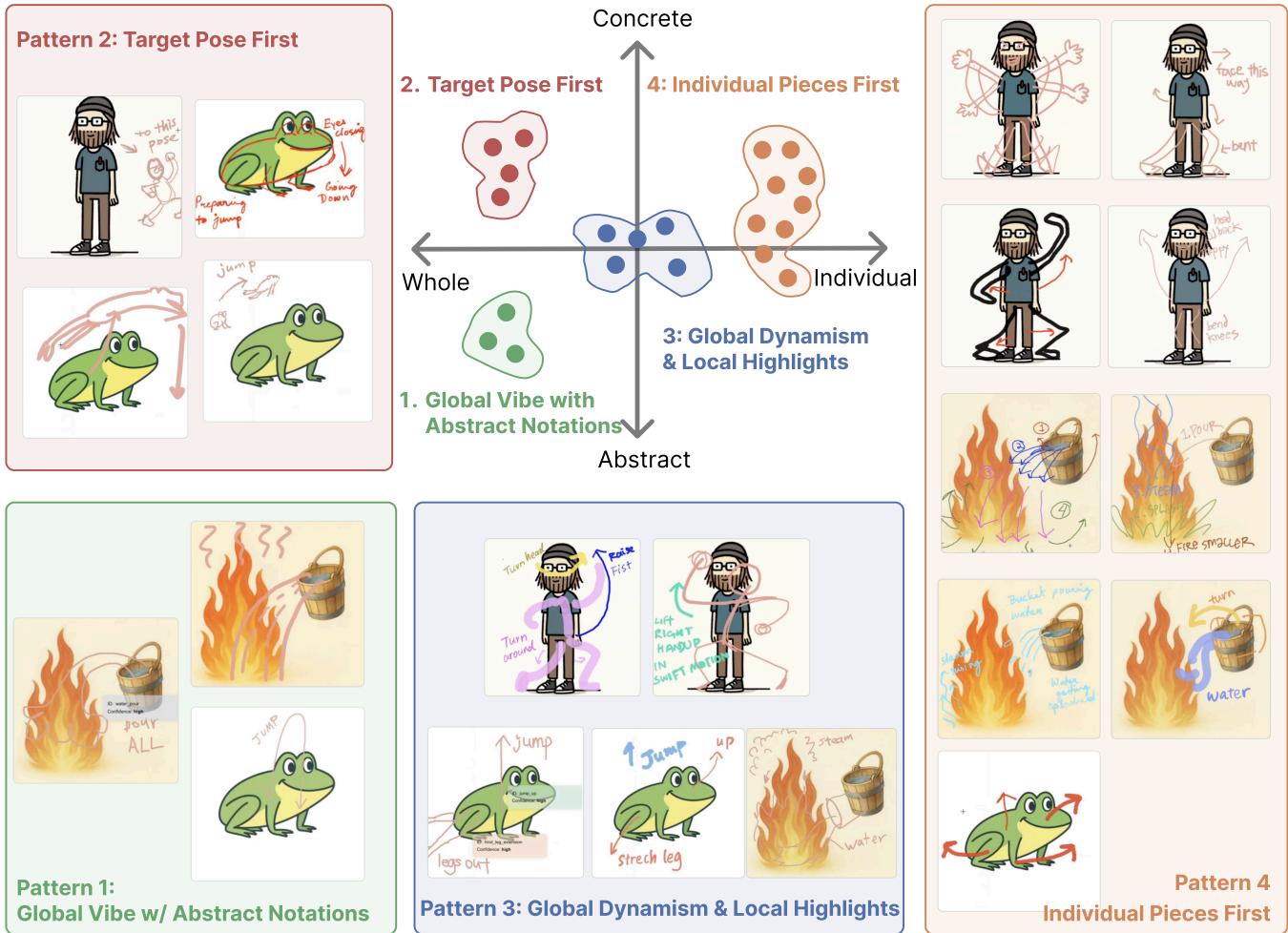


Figure 12: Identified four notation patterns spanning on the axis of the approach (abstract vs.concrete) and the focus (whole vs.individual). Pattern 1 is “Global vibe with abstract notations”, pattern 2 is “Target-pose first”, pattern 3 is “ Global dynamism first, then local highlights”, and pattern 4 is “Individual pieces first”.

down ready for a very high leap, and then hitting the ground again then kind of back up into this main position.”

Pattern 2: Target pose first. This pattern reflects a state-oriented mental model where users focus on *what the next key pose should look like*. They explicitly draw the desired TARGET pose for the whole character (e.g., Figure 11.b), while leaving the transition dynamics (*how* to reach that state from the current one) implicit. This approach reflects a preference for direct artistic execution, as emphasized by P2: “*Let animators draw! ... the overall look is in my mind and naturally comes to my hand.*”

Pattern 3: Global dynamism first, then local highlights. This pattern reflects a hierarchical mental model where users establish the overall motion dynamism before refining local details. They typically begin by notating the overall action using thick strokes, then switch to thinner strokes to annotate specific parts for detailed adjustments (e.g., *bend the knee, turn head*), as shown in Figure 11.c. In other words, a global TARGET for the whole plus selective, part-level constraints to specify the PATH. As P3 explained,

“*that’s actually a big part of animation is you get the central line first and that informs all the rest of your limbs.*”

Pattern 4: Individual pieces first. This pattern reflects a compositional mental model where users decompose complex motion into primitive units and annotate each part separately. Participants leveraged mixed strategies within this pattern, sometimes using abstract notations to specify the PATH for specific limbs, or defining a concrete local TARGET pose for others. They also use color coding to distinguish parts and numerical labels to coordinate timing, e.g., Figure 11.d. However, this granular approach often resulted in visual clutter within the fixed canvas. As P4 noted, “*(when notating,) I felt like restricted in that (fixed) size, being able to zoom in and zoom out might help.*” Although the observed examples typically showed notations for each part appeared together, feedback suggests this was likely an artifact of system latency. P3 explicitly described a desire for an iterative, layered workflow: “*if this worked faster... I would be doing this in several steps... turning his head first and then turning his body... a whole bunch of staggered individual motions.*”

8.2.2 Design implications from notating patterns. Collectively, these four patterns validate the core premise of *notational animating*: rigid vocabularies cannot accommodate the varied mental models of animators, confirming the necessity of *user-defined abstractions*. Beyond this validation, the observed behaviors offer critical design insights regarding the *agency* of AI models and the *structure* of animation representation. Regarding the AI agency, the contrast between the abstract *Global Vibe* (Pattern 1) and the *Target-Pose* (Pattern 2) reveals a dynamic spectrum of control: users shift between offloading low-level planning to the model (acting as a *co-creator* in Pattern 1) and demanding strict adherence to geometric constraints (acting as an *executor* in Pattern 2). Thus, future systems could infer the *intended level of AI agency* from the notation's abstractness.

Furthermore, these notating patterns also point to potential improvements in the *structure* of animation representations. While Pattern 4 (*Individual pieces first*) reflects a compositional structure that aligns with our current representation where all animation units are treated equally, Pattern 3 (*Global dynamism with local highlights*) suggests the need for a hierarchical structure. Future representations could explicitly distinguish between *global* constraints and *local* adjustments so that models can prioritize global dynamism during generation while ensuring that local details complement the primary motion rather than conflict with it.

8.3 Shortcomings in Notations and Models

The observed failures could be attributed to three factors: limitations in notation expressiveness, the robustness of VLM's reasoning, and the instruction-following ability of image generation models.

8.3.1 Expressive affordance of notations. Aligning with our content analysis (Section 3), all participants commented that it was easy to communicate *2D spatial and geometric* information through notation: how the mover should be oriented, where it should move, how it should look (e.g., the pose), how its volume (size) should change (to reflect stretch), and what it should feel like in animation (e.g., squashy and stretchy). However, a fundamental limitation emerged regarding *3D spatial changes* (e.g., twist, tilt, rotate) and *fine-grained occlusion*. Participants (P1, P2, P3, P5, P7) expressed their struggle to flatten complex 3D rotations onto a 2D canvas. It should be noted that this friction lies in specifying *how it rotates in a specific way*, not the *semantic intent of rotation*. This explains why writing text does not help in such cases; articulating a precise 3D trajectory capturing the specific arc, tilt, and velocity of a complex turn is as geometrically ambiguous in natural language as it is in flattened 2D notations. P3 gave an example where it is hard to specify "... *his head moves up while turning around, and then it moves like up and down*." Additionally, *fine-grained local motions with occlusion* (e.g., finger curls) are hard to notate legibly without clutter. P2 noted that "*the marks either became too crowded to read or hard to capture the subtle motions and their dependencies*". P2 suggested that such spatial nuances are best captured not by writing or drawing, but by performance: "*the best way to express this is the gestural input... how I myself perform it*." Finally, *non-geometric transforms* (e.g., color, lighting) are important for animation rhythm and mood but are not currently considered in *notational animating*. P1 also suggested it is possible to indicate *lighting* and *color* with notations.

8.3.2 Interpretation precision and generative fidelity of AI models. In most cases, the vision-language model can interpret users' intention to a reasonable degree; on a 7-point Likert scale evaluating the statement "*The AI correctly understood my intent*.", responses included: 2 agree, 3 somewhat agree, and 2 neutral. However, even when the VLM correctly interprets user intent, the downstream image generation model sometimes fails to execute the instruction faithfully, exhibiting three primary failures: *(i) spatial mismatch*, *(ii) deformation resistance*, and *(iii) stylistic drift*. First, the model sometimes mess up the character's left/right or viewer's left/right, resulting in orientation errors like the unintended flipping in P1's character animation (Figure 13). Second, the current image generator often struggles to apply exaggerated shape deformation (e.g., stretch, compress, overshoot) to subjects it perceives as rigid; for instance, in Figure 13, P6's frog remained a fixed volume despite instructions to compress, whereas P1's cat successfully deformed to some degree. This inconsistency suggests a need to fine-tune models to prioritize explicit deformation instructions over implicit physical realism. Finally, we observed style drift in some cases where the model altered the artistic identity of the base drawing, such as P3's crocodile losing its specific "goofy" aesthetic but being altered to a more generic style.

8.4 Handling Misinterpretation

8.4.1 Understanding AI's interpretation. When evaluating the statement of "*I received sufficient feedback to know where and what to correct when the AI misinterpreted my intent*.", the response included 3 strongly agree, 2 agree, 1 somewhat agree, and 1 neutral. We observed two feedback-checking styles. Most participants adopted a cautious, incremental approach by verifying the system's feedback at each step before committing to full generation; whereas one participant (P2) preferred to ignore intermediate feedback, generate first, and only then inspect the feedback. Overall, participants found the in-place motion tags helpful for quickly revealing how the model parsed their notations. P5 commented on the $\langle \text{source}, \text{path}, \text{target} \rangle$ structure: "... *the first three is enough to give me a quick overview of like what it's saying*." Participants also asked for richer transparency about how secondary notations were interpreted. As P3 suggested, "*If it [the system] could also tell me how it interpreted my different usage of brush sizes and colors, that would be a big help, because if something goes wrong and I know better its thinking process then I know how should I change my notations*."

8.4.2 Correcting the misinterpretations. When the system misinterprets user intent, participants most often corrected it by iterating on their notations. More than half of the participants (4/7) opted to add *secondary notations* as their immediate reaction to misinterpretations, utilizing strategies such as brief text labels to define motion semantics or color-coding to distinguish animation units. The remaining participants (3/7) chose to refine their *primary notations*, for instance, by redrawing arrows or clarifying motion paths. Occasionally, they corrected the interpretation directly by editing motion tags, but this was less preferred, only when iterating on notations keep failing. As P7 commented, "*I don't like switching between drawing and typing*". This explains why most participants chose to iterate on notations rather than editing directly on the

feedback. Finally, when elements were omitted, participants typically fixed this by adding new motions on the timeline blocks to enforce the intended actions.

8.5 Making Exaggerations

Exaggeration distinguishes animation from film; it makes animation believable and often humorous. For an interaction approach for animation keyframe authoring to be truly useful, it must allow animators to easily express and realize exaggeration. In our study, we explored how much *notational animating* currently affords exaggeration across two axes: *motion* and *timing*.

8.5.1 Motion (pose, deformation, range). Exaggeration often lives in dynamic posing and shape deformation; being able to specify and adjust the range, intensity, and shape of motion is crucial. We observed that participants used *secondary notations* such as stroke thickness to signal stronger deformation; for the frog leap, P4 said, “*So stretch will be like that. But I need to be very much. So I need to exaggerate that arrow also, so I’m changing to a larger brush size.*” Participants also frequently utilized dynamic sliders to refine motion range and intensity after the initial generation. On average, each participant engaged with these widgets approximately 8.1 times throughout the session ($MD = 8.0$, $IQR = 3.0$). However, usage frequency varied notably by animation task. Interaction was most frequent during the *open-ended task* ($MD = 3.0$, $IQR = 1.5$) and the *frog leap* task ($MD = 2.0$, $IQR = 1.0$). In contrast, the *fire and water* task saw minimal GUI intervention; two participants (P1 and P6) relied entirely on their notations, while the remaining participants used the sliders only 1.2 times on average ($MD = 1.0$, $IQR = 0.5$). P3 commented, “*I’m happy that I can have that slider for sort of control, animation is all about control to make it real. You know the rate of how this fire diminishes and how far that bucket pours and rotates.*” On the other hand, some participants noted that while dynamic sliders offer quick, high-level adjustments toward an exaggerated feel, they are still less direct than traditional direct manipulation in After Effects; as P6 put it, “*it’s still less direct and less control over things. It’s like suggestion versus direct manipulation.*”

8.5.2 Timing (event order, primary vs. secondary motion). Exaggeration also depends on coordinated timing such as anticipation, follow-through, and overlapping action. As P4 put it, “*things (in animation) actually happen in a natural way. The motion happens with some follow through or some things happen first and resulting to that secondary motion happens. So like if this guy is doing this, he will first bend down and then push, that sort of timing made the exaggerated flavor of animation.*” We observed two strategies: some participants added numeric labels (e.g., “1”, “2”) near marks to indicate event order; others kept the order implicit and expected AI to infer from the context, P2 noted “*it (the system) should know from the physics.*” P6 was impressed when the model inferred anticipation before the leap: “*it was really impressive what it was able to infer of, this is what I was trying to do with the frog balance. Where it does do the compression first before stretching out for the leap as the anticipation. It’s thinking about multiple steps, I think it comes down to a matter of control that is exactly animation needed.*” When AI’s timing inference fell short (e.g., all decomposed motions landing together), participants highlighted the value of incorporating the timeline in *notational*

animating so that they can directly manipulate it to stagger and overlap actions. As P3 commented, “*it’s really impressive I was able to use notations first then down to the timeline because that’s really where you get interesting. Animation is like the overlapping of various actions.*” They also valued per-object tracks, P5 recounted, “*because when you perform an action, multiple stakeholders will be affected and in this case it will be the fire, the water and the bucket itself. So it makes perfect sense to have three different tracks for these separate objects to define.*” Moreover, participants often iterate a lot across notations and the timeline: “*that right feel is not always directly happens, but it happens gradually. You keep on iterating and changing a bit notations and drawing, adjusting the timing, and until the motion and that volume comes up correct.*” -P4

8.6 Mental Model Shifts

We wondered how *notational animating* influences animators’ mental models compared to their daily tools, and found two major shifts in how they think and work. Accordingly, many participants suggested integrating *notational animating* into their current workflow.

8.6.1 From low-level sophisticated control to high-level animation vibes. When asked to compare their prior experience in familiar tools with *notational animating*, all participants described a shift from tweaking parameters toward steering the *vibe* or *feel* of motion at a higher level. P6 described their thinking in *notational animating*, beginning with: “*It’s got squash, it’s got weight, it’s got gravity. It’s talking about these things that I care about.*” P3 contrasted authoring in After Effects with *notational animating* and anticipated a near-term shift toward *notational animating*-style interaction model: “*I know we’re just about that time for change, so hopefully this will go fast. A lot of features that I’ve been using in After Effects lately were designed specifically to do this sort of tweaking around because in After Effects it’s very easy to author keyframes directly. But this way here [notational animating] was introducing a lot of possibilities to kind of like distribute those key frames over an amount of time or sometimes distribute the values over an amount of time so that people can kind of just grab on to the totality and the vibe of the animation instead of stacking all stuff manually.*” -P5, an independent game designer who often codes animations, reflected: “*when I program it in a loop, I’m just moving X-Y coordinates. Annotation feels more direct for me because I’m literally just put down the motion and the feel and all what I want with the pen. It helps me think and feel it intuitively, compared to that variables.*” P7 commented that, “*sort of like blending the brainstorming sketching phase in the animation phase. It’s just like you’re kind of vibing it out.*”

8.6.2 From doing “one thing at a time” to “multiple things together”. Animators in classical workflow typically do “one thing at a time”: “*we normally do an animation where all these things are different pieces and animate one at a time, test it until it works, then move on to the next, then stagger them together.*” -P6 However, in our study, we observed participants often choose to animate multiple things together. In the water and fire example, we observed most animators *notate* bucket, water, fire, and steam altogether and leave AI to *infer* the timing and their dependencies. P1 explained that “*there’s some interconnection between these things. These are all kind of they’re coming together as a system. So that’s pretty cool that I was able to*

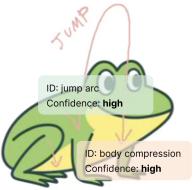
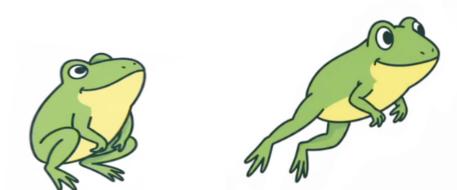
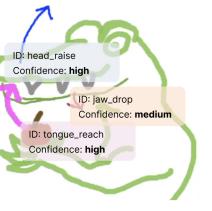
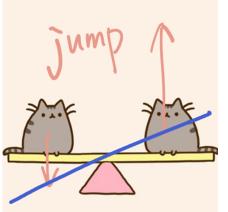
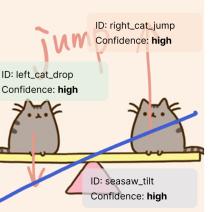
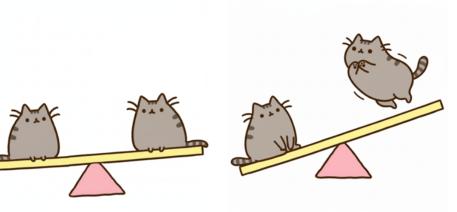
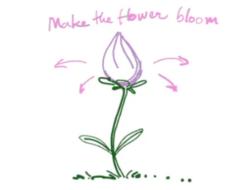
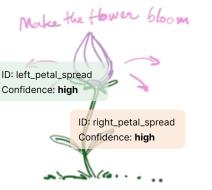
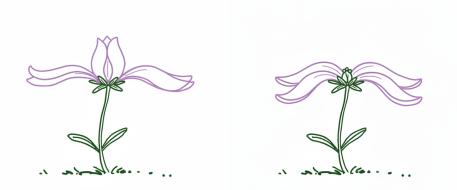
	Notation	Interpretation	Generated Key Frames
P1			
P7			
P6			
P3			
P1			
P4			

Figure 13: Participant outputs from the expert evaluation: the first three results are from the structured task; the last three were produced during the open-ended exploratory task.

work on them altogether.” Similarly, in the frog example, P6 used one notation to indicate the whole process that the frog first compress to prepare the jump then jump a big leap arc. They reflected that “*I think of it all as one thing and all together. I love that I was able to tell a specific order of events, like use that little down thing and the big arc to specify the anticipation at the beginning before the jump, and I’m impressed that it was able to figure that out. I’m really interested to use it to see how multi-frame animation comes together.*”

8.6.3 Integrating notational animating into current workflow. Participants showed great interests in getting notational input plug into established pipelines to handle both high-level “vibe-first” guidance and low-level precise edits. On the one hand, P6 envisioned a hybrid model inside After Effects: “*bringing this kind of system into AE feels like the best of both worlds because it still gives me the full control and precision I’m already used to. Wouldn’t it be cool if there were this additional things that could let me quickly try things out and potentially fill in some of the gaps of things I don’t know how to do as well.*” On the other hand, besides rough and quick experiments, P2 also suggested “*notations can also be used for very minor adjustments*”. As P4 noted, “*it can be useful when they just want to tweak a final animation they have created, like okay, this keyframe here I just want this to move a bit here*”. Participants also anticipated different adoption curves. P4 cautioned that, for professionals, “*that might take some time to totally get into their pipeline. All the professional who are already in the industry, they are very accustomed to their own workflows, the changing will be gradual; but this will be quite interesting with the new generation of GenAI artists who are trying out new ways of telling story and motion.*”

9 Discussion

We regard this paper as an initial step towards realizing *notational animating*. Overall, our qualitative findings show that the core interaction is intuitive and useful for animators: it draws on their familiar, conventional sketching and requires no learning. With our prototype, participants were able to express a range of animation nuances, including multiple forms of exaggeration, and were often impressed that most informal notations can be correctly interpreted into their intended animation. Collectively, these results support *notational animating* as a promising paradigm for higher-level animation keyframe authoring which has the potential to serve as a complement (but not a replacement) to classic GUIs (e.g., After Effects), which excel at fine-grained, low-level precision control.

9.1 Reflections and Future Directions

Despite participants feeling they can express most of their intent with *notational animating*, there is still much room for improvements. We see the opportunities and needs for expanding the notations. For example, camera movement is conventionally expressed through marks, and prior work has examined arrow-based notations for camera motion in storyboarding. Participants also suggested extending notation to non-geometric transformations (e.g., color and lighting). Accommodating these additions will require extending our current formal representation of animation intent. We designed this representation to be extensible so new dimensions can be accommodated without breaking the existing structure.

We also identified some bottlenecks related to current models to make *notational animating* more useful. First, the intention inference and keyframe generation should be faster. A tight feedback loop encourages bold exploration; high latency will push users toward conservative choices. Moreover, what matters for professional adoption ultimately is the fidelity of the generated keyframes. Although optimizing output quality was not our focus here, it is decisive for whether *notational animating* can shoulder production work or remains primarily an ideation tool for quick, rough exploration. We are optimistic that near-term model advances will be able to make progress on interpretation robustness and generative fidelity (as discussed in Section 8.3.2) to enable *notational animating* to mature into a serious production approach. Additionally, our current framework lacks a dedicated mechanism to prevent the VLM from imposing unsolicited creative intent when users require strict control. As discussed in Section 8.2.2, the abstractness of a user’s notation serves as a strong signal of their intended agency. Future systems should leverage this signal to dynamically regulate the model’s behavior, specifically inhibiting the tendency to “polish” or restyle the animation when users demonstrate a preference for explicit, low-level control. We hope this work can help bridge the community of HCI and AI, calling out the needs for understanding artists’ vision such as principles (e.g., anticipation, follow-through, exaggeration) which go beyond real-world physics, so models can reason about and generate the intended animation faithfully.

Beyond keyframes, the “last mile” we envision is automatically rendering in-between frames from the produced keyframes (most likely via video models). Although it is feasible to do so with the state-of-the-art models, results are not yet production-ready, and current video generation remains slow. Until generation is fast enough for interactive playback during design, animators cannot easily use full animated previews to test and refine their choices. Nevertheless, we remain optimistic that rapid advances in video models will make this practical in the near future.

9.2 Animation Representations across Different Abstraction Levels

Central to our *notational animating* is the formalized animation intention representation. If we look back to the history of the animation creation, each interaction model corresponds to a representation for animation. Back to the classical hand-drawing era, animation representation was abstract, artist-centric constructs in simplest visual form (e.g., line of action) where artists start from the simplest forms and get the dynamism of animation and iteratively add details on top of it to get the final animation. Later in the digital animation era, the representation shifted to concrete, lower-level primitives: per-keyframe translation, rotation, scaling, and physical simulations. Interestingly, this sharp change of the representation from the abstract to the concrete one is not really aligned with how artists think about the animation, even today, master animators still tend to think abstractly. This separation likely explains why animation production stages are distinct, from initial freeform sketches to final production software [23].

Recently, with AI’s advancements, it is promising to offload more low-level tedious labor to AI, and our focus in this paper is to facilitate a direct manipulation interaction that accommodates different

levels of abstraction of their intent. We want to highlight the way in *notational animating* that adding a middle-layer representation (our formalized animation intention representation) bridges the fluid, sketch-like notations that artists prefer and the implicit space AI works on. As a formal layer between human-facing notation and AI embeddings, it aims to be extensible and flexible, aligning abstract intent with concrete execution.

Our goal is not to define a single best representation for AI-driven animation authoring. Instead, we aim to open a design space and invite the community to explore feasible higher-level representations that align with how animators think and work. We approach the problem from a *movement-centric* perspective, i.e., “what aspects about movements users wish to specify”; but other ways such as *effect-centric* representation are also promising, e.g., reifying animation principles or physics-driven effects as first-class instruments [27, 56]. Style is also central to animation (e.g., manga, Pixar, Disney). Future work could let users define and reuse personal style conventions as part of their notations. More broadly, it remains open how such notations evolve over time with cultural practices. We hope *notational animating* is a step toward representations that are both human-legible and AI-operable, blurring the boundary between ideation and execution in animation authoring.

9.3 Limitations

Our study recruited only professional animators. This study design prioritized understanding how *notational animating* fits into professional practice and its *expressive ceiling* in a domain governed by nuanced principles (e.g., anticipation, follow-through, exaggeration). As such, experts are better positioned to set demanding goals, articulate complex intentions, and judge whether this interaction approach helps make motion “a certain way” rather than merely making things move. Their proficiency with GUI-based tools (e.g., After Effects) also enabled sharper contrasts between interaction paradigms. We anticipate that *notational animating* will also benefit novices, though their perception and usage patterns may differ from those of professionals. Professional animators possess internalized knowledge of implicit notation conventions gained through training and practice. In contrast, novices lacking this exposure may face a *cold start* problem, unsure of how to initiate the interaction. A possible solution is to offer a set of *default* notations or generating context-aware ambient suggestions. These aids would help novices discover *what* can be notated and *how*, eventually preparing them to invent their own notations. Evaluating the learning curves of novices remains a critical direction for future work.

Our current evaluation is qualitative in nature, focusing on understanding how professionals utilize *notational animating* within our prototype. Future quantitative measurements could deepen our understanding of *notational animating*. Although *notational animating* is grounded in real-world sketching practice, it remains unclear how closely users’ digital notations match what they would sketch on paper. We hypothesize that notation behaviors shift with both the medium (static paper vs. an interactive system with feedback) and the audience (self-reflection vs. communicating with an AI model). To test this, a comparative study could ask participants to express the same animation intentions using both paper and *notational animating*, enabling us to quantify systematic differences in

notation. Finally, real animation projects often span days to months. Although our in-lab study included an open-ended exploration session that let participants experiment with their own drawings and any effects they wished, the tasks remained toy-scale relative to production work. A longitudinal deployment study on real projects will therefore be invaluable for assessing the broader, real-world impact of *notational animating*.

10 Conclusion

In this paper, we introduce *notational animating*, an interaction concept for animation keyframe authoring that reifies sketched notations as animators’ intentions. These notations guide the creation and editing of keyframes, which can subsequently be consumed by automatic methods (e.g., recent frame-interpolation models) to generate animation frames. Through content analysis and expert interviews, we characterize the notations’ informal nature: non-categorical, ambiguous, and combinatorial. We then formalize them into a structured representation of animation intent, enabling recent vision-language models to interpret the notations systematically. As an initial step toward realizing *notational animating*, we built a prototype featuring a two-level feedback mechanism and dynamic UI widgets that provide fine-grained control complementing the high-level notations. In a preliminary expert evaluation, we examined how animators perceive and use *notational animating*. Our findings suggest that sketched notations can effectively support animation authoring and inform future research on AI-assisted animation tools that preserve artists’ agency.

Acknowledgments

We sincerely thank Seth Walker, Val Head, and Wilmot Li for their insightful discussions during the early stages of this project, and the anonymous reviewers for their invaluable suggestions, which greatly improved this work. We also thank Crystina Zhang for her kind help with several figures. Finally, we are grateful to all our participants for their time and valuable input.

This work is supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant (#RGPIN-2020-03966), the Canada Foundation for Innovation (CFI) John R. Evans Leaders Fund (JELF) (#42371), and a gift fund from Adobe. We acknowledge that much of our work takes place on the traditional territory of the Neutral, Anishinaabeg, and Haudenosaunee peoples. Our main campus is located on the Haldimand Tract, the land granted to the Six Nations that includes six miles on each side of the Grand River.

References

- [1] Jessica Abel and Matt Madden. 2008. *Drawing words and writing pictures: Making comics: Manga, graphic novels, and beyond*. Vol. 1. Macmillan.
- [2] Malihe Alikhani and Matthew Stone. 2018. Arrows are the Verbs of Diagrams. In *Proceedings of the 27th International Conference on Computational Linguistics*, Emily M. Bender, Leon Derczynski, and Pierre Isabelle (Eds.). Association for Computational Linguistics, Santa Fe, New Mexico, USA, 3552–3563. <https://aclanthology.org/C18-1301/>
- [3] Ian Arawjo, Anthony DeArmas, Michael Roberts, Shrutarshi Basu, and Tapan Parikh. 2022. Notational Programming for Notebook Environments: A Case Study with Quantum Circuits. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology* (Bend, OR, USA) (*UIST '22*). Association for Computing Machinery, New York, NY, USA, Article 62, 20 pages. doi:10.1145/3526113.3545619

[4] Omri Avrahami, Ohad Fried, and Dani Lischinski. 2023. Blended latent diffusion. *ACM Transactions on Graphics (TOG)* 42, 4 (2023), 1–11.

[5] Michel Beaudouin-Lafon. 2000. Instrumental interaction: an interaction model for designing post-WIMP user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 446–453.

[6] Samuelle Bourgault, Li-Yi Wei, Jennifer Jacobs, and Rubaiat Habib Kazi. 2025. Narrative Motion Blocks: Combining Direct Manipulation and Natural Language Interactions for Animation Creation. In *Proceedings of the 2025 ACM Designing Interactive Systems Conference (DIS '25)*. Association for Computing Machinery, New York, NY, USA, 1366–1386. doi:10.1145/3715336.3735766

[7] Xiang 'Anthony' Chen, Jeff Burke, Ruofei Du, Matthew K Hong, Jennifer Jacobs, Philippe Laban, Dingzeyu Li, Nanyun Peng, Karl DD Willis, Chien-Sheng Wu, et al. 2023. Next steps for human-centered generative ai: A technical perspective. *arXiv preprint arXiv:2306.15774* (2023).

[8] John Joon Young Chung and Eytan Adar. 2023. PromptPaint: Steering Text-to-Image Generation Through Paint Medium-like Interactions. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. 1–17.

[9] John Joon Young Chung, Wooseok Kim, Kang Min Yoo, Hwaran Lee, Eytan Adar, and Minsuk Chang. 2022. TaleBrush: Sketching stories with generative pretrained language models. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–19.

[10] Marianela Ciolfi Felice, Nolwenn Maudet, Wendy E Mackay, and Michel Beaudouin-Lafon. 2016. Beyond snapping: Persistent, tweakable alignment and distribution with sticky lines. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. 133–144.

[11] Neil Cohn. 2013. *The Visual Language of Comics: Introduction to the Structure and Cognition of Sequential Images*. A&C Black.

[12] Neil Cohn. 2020. Visual narrative comprehension: Universal or not? *Psychonomic Bulletin & Review* 27, 2 (2020), 266–285.

[13] Neil Cohn and Stephen Maher. 2015. The notion of the motion: The neurocognition of motion lines in visual narratives. *Brain research* 1601 (2015), 73–84.

[14] Richard C Davis, Brien Colwell, and James A Landay. 2008. K-sketch: a'kinetic'sketch pad for novice animators. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 413–422.

[15] Will Eisner. 2008. *Comics and sequential art: Principles and practices from the legendary cartoonist*. WW Norton & Company.

[16] Yu Gao, Haoyuan Guo, Tuyen Hoang, Weilin Huang, Lu Jiang, Fangyuan Kong, Huixia Li, Jia Shi Li, Liang Li, Xiaojie Li, Xunsong Li, Yifu Li, Shanchuan Lin, Zhiye Lin, Jiawei Liu, Shu Liu, Xiaonan Nie, Zhiwu Qing, Yuxi Ren, Li Sun, Zhi Tian, Rui Wang, Sen Wang, Guoqiang Wei, Guohong Wu, Jie Wu, Ruiqi Xia, Fei Xiao, Xuefeng Xiao, Jiangqiao Yan, Ceyuan Yang, Jianchao Yang, Runkai Yang, Tao Yang, Yihang Yang, Zilyu Ye, Xuejiao Zeng, Yan Zeng, Heng Zhang, Yang Zhao, Xiaozheng Zheng, Peihao Zhu, Jiaxin Zou, and Feilong Zuo. 2025. Seedance 1.0: Exploring the Boundaries of Video Generation Models. *arXiv:2506.09113* (2025). <https://arxiv.org/abs/2506.09113>

[17] Joseph Gilland. 2012. *Elemental Magic, Volume 2: The Technique of Special Effects Animation*. Routledge.

[18] Thomas RG Green. 1989. Cognitive dimensions of notations. *People and computers V* (1989), 443–460.

[19] Martin Guay, Marie-Paule Cani, and Rémi Ronfard. 2013. The line of action: an intuitive interface for expressive character posing. *ACM Trans. Graph.* 32, 6, Article 205 (Nov. 2013), 8 pages. doi:10.1145/2508363.2508397

[20] Martin Guay, Rémi Ronfard, Michael Gleicher, and Marie-Paule Cani. 2015. Space-time sketching of character animation. *ACM Transactions on Graphics (ToG)* 34, 4 (2015), 1–10.

[21] Edwin L Hutchins, James D Hollan, and Donald A Norman. 1986. Direct manipulation interfaces. In *User centered system design*. CRC Press, 87–124.

[22] Takeo Igarashi, Tomer Moscovitch, and John F Hughes. 2005. As-rigid-as-possible shape manipulation. *ACM transactions on Graphics (TOG)* 24, 3 (2005), 1134–1141.

[23] Amir Jahanlou, William Odorn, and Parmit Chilana. 2021. Challenges in getting started in motion graphic design: Perspectives from casual and professional motion designers. In *Graphics Interface 2021*.

[24] Avneet Kaur, Hubert Leo, Nachiket Pujari, and Bret Bays. 2025. Choreography of Hair and Cloth in Disney's "Moana 2". In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Talks (SIGGRAPH Talks '25)*. Association for Computing Machinery, New York, NY, USA, Article 13, 2 pages. doi:10.1145/3721239.3734105

[25] Rubaiat Habib Kazi, Fanny Chevalier, Tovi Grossman, and George Fitzmaurice. 2014. Kitty: sketching dynamic and interactive illustrations. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology* (Honolulu, Hawaii, USA) (UIST '14). Association for Computing Machinery, New York, NY, USA, 395–405. doi:10.1145/2642918.2647375

[26] Rubaiat Habib Kazi, Fanny Chevalier, Tovi Grossman, Shengdong Zhao, and George Fitzmaurice. 2014. Draco: bringing life to illustrations with kinetic textures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada) (CHI '14). Association for Computing Machinery, New York, NY, USA, 351–360. doi:10.1145/2556288.2556987

[27] Rubaiat Habib Kazi, Tovi Grossman, Nobuyuki Umetani, and George Fitzmaurice. 2016. Motion Amplifiers: Sketching Dynamic Illustrations Using the Principles of 2D Animation. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) (CHI '16). Association for Computing Machinery, New York, NY, USA, 4599–4609. doi:10.1145/2858036.2858386

[28] Joonho Kim and Karan Singh. 2024. Squidgets: Sketch-based Widget Design and Direct Manipulation of 3D Scene. *arXiv preprint arXiv:2402.06795* (2024).

[29] Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. 2024. Hunyuanvideo: A systematic framework for large video generative models. *arXiv:2412.03603* (2024).

[30] Vivian Liu, Rubaiat Habib Kazi, Li-Yi Wei, Matthew Fisher, Timothy Langlois, Seth Walker, and Lydia Chilton. 2025. LogoMotion: Visually-Grounded Code Synthesis for Creating and Editing Animation. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) (CHI '25). Association for Computing Machinery, New York, NY, USA, Article 157, 16 pages. doi:10.1145/3706598.3714155

[31] Jiaju Ma and Maneesh Agrawala. 2025. MoVer: Motion Verification for Motion Graphics Animations. *ACM Trans. Graph.* 44, 4, Article 33 (July 2025), 17 pages. doi:10.1145/3731209

[32] Damien Masson, Sylvain Malacia, Géry Casiez, and Daniel Vogel. 2024. Direct-GPT: A Direct Manipulation Interface to Interact with Large Language Models. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) (CHI '24). Association for Computing Machinery, New York, NY, USA, Article 975, 16 pages. doi:10.1145/3613904.3642462

[33] Scott McCloud. 2006. Making comics: Storytelling secrets of comics, manga and graphic novels. (2006).

[34] Scott McCloud and Mark Martin. 1993. *Understanding comics: The invisible art*. Vol. 106. Kitchen sink press Northampton, MA.

[35] Xingang Pan, Ayush Tewari, Thomas Leimkühler, Lingjie Liu, Abhimitra Meka, and Christian Theobalt. 2023. Drag your gan: Interactive point-based manipulation on the generative image manifold. In *ACM SIGGRAPH 2023 Conference Proceedings*. 1–11.

[36] Ryan Po, Yotam Nitzan, Richard Zhang, Berlin Chen, Tri Dao, Eli Shechtman, Gordon Wetzstein, and Xun Huang. 2025. Long-Context State-Space Video World Models. *arXiv:2505.20171* (2025). <https://arxiv.org/abs/2505.20171>

[37] Nathalie Riche, Anna Offenwanger, Frederic Gmeiner, David Brown, Hugo Romat, Michel Pahud, Nicolai Marquardt, Kori Inkpen, and Ken Hinckley. 2025. AI-Instruments: Embodiment Prompts as Instruments to Abstract & Reflect Graphical Interface Commands as General-Purpose Tools. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems* (CHI '25). Association for Computing Machinery, New York, NY, USA, Article 1104, 18 pages. doi:10.1145/3706598.3714259

[38] Stephanie Santos, Fanny Chevalier, Ravin Balakrishnan, and Karan Singh. 2013. Direct space-time trajectory control for visual media editing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Paris, France) (CHI '13). Association for Computing Machinery, New York, NY, USA, 1149–1158. doi:10.1145/2470654.2466148

[39] Vishnu Sarukkai, Lu Yuan, Mia Tang, Maneesh Agrawala, and Kayvon Fatahalian. 2024. Block and Detail: Scaffolding Sketch-to-Image Generation. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology* (Pittsburgh, PA, USA) (UIST '24). Association for Computing Machinery, New York, NY, USA, Article 33, 13 pages. doi:10.1145/3654777.3676444

[40] Xinyu Shi, Yinghou Wang, Ryan Rossi, and Jian Zhao. 2025. Brickify: Enabling Expressive Design Intent Specification through Direct Manipulation on Design Tokens. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems* (CHI '25). Association for Computing Machinery, New York, NY, USA, Article 424, 20 pages. doi:10.1145/3706598.3714087

[41] Xinyu Shi, Yinghou Wang, Yun Wang, and Jian Zhao. 2024. Piet: Facilitating Color Authoring for Motion Graphics Video. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) (CHI '24). Association for Computing Machinery, New York, NY, USA, Article 148, 17 pages. doi:10.1145/3613904.3642711

[42] Aude Simkute, Lev Tankelevitch, Viktor Kewenig, Ava Elizabeth Scott, Abigail Sellen, and Sean Rintel. 2024. Ironies of Generative AI: Understanding and Mitigating Productivity Loss in Human-AI Interaction. *International Journal of Human-Computer Interaction* (2024), 1–22.

[43] Harrison Jesse Smith, Qingyuan Zheng, Yifei Li, Somya Jain, and Jessica K. Hodges. 2023. A Method for Animating Children's Drawings of the Human Figure. *ACM Trans. Graph.* 42, 3, Article 32 (June 2023), 15 pages. doi:10.1145/3592788

[44] Hari Subramonyam, Roy Pea, Christopher Pondoc, Maneesh Agrawala, and Colleen Seifert. 2024. Bridging the Gulf of Envisioning: Cognitive Challenges in Prompt Based Interactions with LLMs. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 1–19.

[45] Lev Tankelevitch, Viktor Kewenig, Aude Simkute, Ava Elizabeth Scott, Advait Sarkar, Abigail Sellen, and Sean Rintel. 2024. The metacognitive demands and opportunities of generative AI. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 1–24.

- [46] Maham Tanveer, Yang Zhou, Simon Niklaus, Ali Mahdavi Amiri, Hao Zhang, Krishna Kumar Singh, and Nanxuan Zhao. 2024. Motionbridge: Dynamic video inbetweening with flexible controls. *arXiv preprint arXiv:2412.13190* (2024).
- [47] Frank Thomas and Ollie Johnston. 1981. *Disney Animation: The Illusion of Life*. Abbeville Press.
- [48] Matthew Thorne, David Burke, and Michiel Van De Panne. 2004. Motion doodles: an interface for sketching character motion. *ACM Transactions on Graphics (ToG)* 23, 3 (2004), 424–431.
- [49] Tiffany Tseng, Ruijia Cheng, and Jeffrey Nichols. 2024. Keyframer: Empowering animation design using large language models. *arXiv preprint arXiv:2402.06071* (2024).
- [50] Priyan Vaithilingam, Elena L Glassman, Jeevana Priya Inala, and Chenglong Wang. 2024. DynaVis: Dynamically Synthesized UI Widgets for Visualization Editing. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 1–17.
- [51] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwei Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. 2025. Wan: Open and Advanced Large-Scale Video Generative Models. *arXiv:2503.20314* (2025). <https://arxiv.org/abs/2503.20314>
- [52] Colin Ware. 2010. *Visual thinking for design*. Elsevier.
- [53] Justin D Weisz, Jessica He, Michael Muller, Gabriela Hoefer, Rachel Miles, and Werner Geyer. 2024. Design Principles for Generative AI Applications. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 1–22.
- [54] Richard Williams. 2009. *The Animator's Survival Kit-Revised Edition: A Manual of Methods, Principles and Formulas for Classical, Computer, Games, Stop Motion and Internet Animators*. Faber & Faber, Inc.
- [55] Zizhao Wu, Qin Wang, Xinyang Zheng, Jianglei Ye, Ping Yang, Yunhai Wang, and Yigang Wang. 2025. Doodle Your Motion: Sketch-Guided Human Motion Generation. *IEEE Transactions on Visualization and Computer Graphics* 31, 9 (2025), 6538–6549. doi:10.1109/TVCG.2024.3521333
- [56] Jun Xing, Rubaia Habib Kazi, Tovi Grossman, Li-Yi Wei, Jos Stam, and George Fitzmaurice. 2016. Energy-Brushes: Interactive Tools for Illustrating Stylized Elemental Dynamics. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology* (Tokyo, Japan) (UIST '16). Association for Computing Machinery, New York, NY, USA, 755–766. doi:10.1145/2984511.2984585
- [57] Ryan Yen, Jian Zhao, and Daniel Vogel. 2025. Code Shaping: Iterative Code Editing with Free-form AI-Interpreted Sketching. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI '25)*. Association for Computing Machinery, New York, NY, USA, Article 872, 17 pages. doi:10.1145/3706598.3713822
- [58] Richard Zhang, Jun-Yan Zhu, Phillip Isola, Xinyang Geng, Angela S Lin, Tianhe Yu, and Alexei A Efros. 2017. Real-time user-guided image colorization with learned deep priors. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 1–11.
- [59] Sharon Zhang, Jiajia Ma, Jiajun Wu, Daniel Ritchie, and Maneesh Agrawala. 2023. Editing Motion Graphics Video via Motion Vectorization and Transformation. *ACM Trans. Graph.* 42, 6, Article 229 (Dec. 2023), 13 pages. doi:10.1145/3618316
- [60] Lei Zhong, Chuan Guo, Yiming Xie, Jiawei Wang, and Changjian Li. 2025. Sketch2Anim: Towards Transferring Sketch Storyboards into 3D Animation. *ACM Trans. Graph.* 44, 4, Article 74 (July 2025), 15 pages. doi:10.1145/3731167

Appendix

A Prompt for Interpreting Notations

Prompt for Interpreting Notations:

Role & Goal:

Interpret raster sketches with motion marks. For each mark, identify its role, group marks or decompose a mark (only when necessary) into motion units, and return a structured description using <source, path, target> plus relevant secondary modifiers.

Conceptual Model:

- Motion unit: one intended motion.
- Primary tuple <Source, Path, Target>
 - Source: the moving part(s).
 - Path: movement type and direction.
 - Target: intended end state.
- Secondary modifiers: visual properties, text/letters/numbers, emphasis, symbols.
- Prioritize role over form. Allow incomplete units; infer missing pieces when intent is clear.

Grouping & Association:

- Cluster by proximity, common style (color/thickness/texture), and shared role/direction.
- Associate nearby numbers/letters/text with the most relevant path or target.

Grid:

- 30x30 cells; origin at bottom-left (0,0); x-> right, y-> up.
- Use integer intersections 0-30 and .5 for edges/centers.

What to Extract (per motion unit)

- color: unique per unit.
- roi_bbox: [x_min, y_min, x_max, y_max] in grid units.
- primary: { source, path, target } – concise, consistent with the model above.
- secondary_modifiers: entries of { property: color|thickness|text|number|letter|style|other; value: "..."; intended_meaning: "..."; scope: source|path|target|unit }.
- temporal_order: integer if deducible from nearby numbers/letters; else null.
- confidence: 0-1.
- natural_language_summary: 1-3 sentences describing the intended move and end state (where it lands, contacts/near-contacts, rough orientation/pose); include 2-4 feel adjectives (e.g., fluid, snappy, heavy, brisk).
- unassigned_marks: brief notes with small grid bboxes for marks not confidently assigned.
- global_timeline: ordered list of unit ids if numbers/letters imply a sequence.
- legend_inferred: propose consistent color/style meanings if patterns emerge (e.g., "red = paths").

Dimensionality Controls (required)

Provide 1-3 neutral sliders, default 1.0, grounded in marks:

- Amplitude/geometry (pick 0-2): e.g., height of raise, reach length, arc sweep amount.
- Directional bias (<= 1): e.g., forward bias of the swing. (0.5-1.5)
- Timing/Energy (when cued by echoes/weight): e.g., tempo/quickness.

Output Format:

Return only a single valid JSON object containing the above keys.

B Coding Scheme

Table 3 summarizes the coding scheme we used to analyze the 135 collected notation sketches. The scheme is organized by the three functional roles identified in our content analysis: *geometric guides*, *spatial guides*, and *motion amplifiers*. We further subdivides each role into sub-categories (e.g., force, path, pose, style) with concrete code labels and brief operational definitions. During coding, annotators could assign any number of codes to a single drawing.

C Notation Analysis Details

Table 4 provides an index of the 135 collected notation sketches, linking each example to its source and the codes assigned during our notation analysis. For each sketch, we provide its URL, along with the corresponding category, sub-category, and code label(s) from our coding scheme.

D VLM Responses with and without Structured Representation

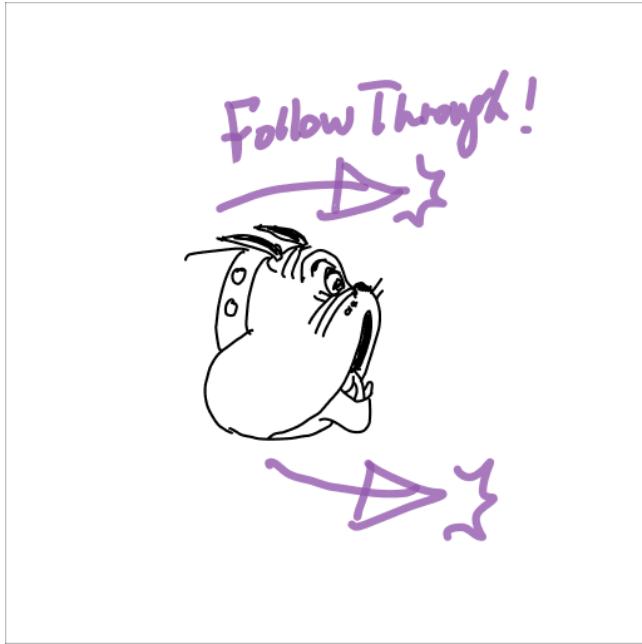


Figure 14: The test case to interpret with and w/o the structured representation.

In this section, we present a qualitative comparison between VLM responses generated with and without our structured representation, interpreting the case of Figure 14. While the unstructured baseline correctly identifies the general context of a head movement, it fails to capture the key essence of how the animation principle, follow through, applies to the soft ear and cheeks of the character instead of the whole head of the character. Specifically, it aggregates the distinct movements of the ears and jowls into a single “gentle S-curve” trajectory, treating the head as a rigid body. It overlooks the critical temporal and spatial separation between the loose upper

and lower masses, losing the nuance of the overlapping action that defines the follow-through effect.

VLM's response w/o structured representation

The sketch is giving a note for “follow-through” on the bulldog’s head move. The bulldog snaps its head toward the right. After the main snap, the head should keep traveling along the same right-ward arc, dipping slightly downward, then settle. The path of action is indicated by two curved arrows that form a gentle S-curve – first a small upward-right swing, then a broader downward-right sweep. Little star-shaped burst marks at the arrow tips suggest a soft ease-out / settling at the end of the arc.

VLM's response with structured representation

• UNIT 1

– Primary Components

- * **Source:** Loose upper portions of the dog’s head (ear tips / top-tuft).
- * **Path:** Broad, slightly upward-curving arrow that sweeps left-to-right over the head.
- * **Target:** Positions a little farther to camera-right, continuing after the main head stop (follow-through).

– Secondary Modifiers

- * **Color:** Purple (*Meaning: motion cue, Scope: unit*)
- * **Thickness:** Medium (*Meaning: principal path, Scope: path*)
- * **Text:** “Follow Through!” (*Meaning: clarify that the motion is a follow-through action, Scope: unit*)
- * **Symbol:** Small 4-point star at arrow tip (*Meaning: accent / settle point, Scope: target*)

– Temporal Order: 1

- **Natural Language Summary:** After the head stops, the floppy ear and top mass keep sliding a little farther to the right, illustrating the follow-through principle.

• UNIT 2

– Primary Components

- * **Source:** Loose lower portions of the head (cheeks/jowls & collar tag).
- * **Path:** Gentle left-to-right, slightly downward arrow arcing beneath the chin.
- * **Target:** Overshoots to a lowered, right-shifted settle position.

– Secondary Modifiers

- * **Color:** Purple (*Meaning: motion cue, Scope: unit*)
- * **Thickness:** Medium (*Meaning: principal path, Scope: path*)
- * **Symbol:** Small 4-point star at arrow tip (*Meaning: accent / settle point, Scope: target*)

– Temporal Order: 2

- **Natural Language Summary:** Simultaneously or a beat later, the heavy jowls and collar tag lag and then swing past the stopping point of the skull, settling lower and farther to the right.

Table 3: The Coding scheme used for notation analysis for the 135 real-world collected drawings. During the coding, coders can assign any number of codes from this table to the same drawing.

Category	Sub-category	Code Label	Definition
Force (animation principles)	Overshoot	Overshoot	Any marks indicating the object travels past its target destination before settling back.
		Anticipation	Any marks indicating a preparatory action in the opposite direction of the main action.
	Drag / Follow through	Drag / Follow through	Any marks on/near the passive and loose parts (e.g., hair, cloth, tails), showing them lagging behind (drag) or swinging forward (follow through) the main movement.
	Deformation (stretch/squash)	Deformation (stretch/squash)	Any marks that distort the shape to show flexibility: making the object long and thin for stretching, short and wide for squashing.
Force (actual forces)	Deformation (twist/tilt/bend)	Deformation (twist/tilt/bend)	Any marks indicating the rotation of a body part around its own axis (e.g., hips, spine, neck) or bending of the subject, often for storing the energy.
		Gravity / Weight	Any marks (often vertical lines, arrows pointing down) indicating the downward pull or heaviness of an object .
	Contact (Friction/Push off)	Contact (Friction/Push off)	Any marks at/near the point of interaction with a surface (e.g., feet on floor) indicating frictions, launch force, or bounces .
		Wind / Air flow	Any marks surrounding the subject indicating wind direction .
Force (action-specific)	Strike Force (e.g., punch, hit, kick)	Strike Force (e.g., punch, hit, kick)	Any marks directing a strike, punch, hit, usually straight or sharp .
		Rotational Force (e.g., swing / throw)	Any marks indicating the curved path of an object traveling through space (e.g., a swing, a throw, a wide turn). Exclude if the object stays in roughly the same location; distinguish from twist/tilt.
	Action Vector (e.g., lift up)	Action Vector (e.g., lift up)	Linear arrows indicating the direction of internal force, such as muscular effort (e.g., the “Lift up”, “push off”).
		Line of action	Broad, sweeping primary curve(s) on the subject that defines the overall posture, axis, and energy of the pose .
Geometric guides	Pose (living)	Pose contour	Multiple marks around the subject to outline the contour of the subject , often used to show how the subject’s outer shape look like.
		Volumetric primitives	Marks represent volumetric primitives (boxes, circles, triangles, etc.) to show the scale of a specific part of the body .
	Pose (non-living)	Fabric folds	Marks defining the stress points, tension, pinching, or bunching of clothing/material .
		Elemental form	General boundary shapes or flow lines guiding groups of small amorphous elements (smoke, fire, water, dust).
Path	Trajectory (untimed)	Trajectory (untimed)	A line or multiple continuous marks tracing the future path of movement .
		Trajectory (timed)	A path line that includes small hash marks (ticks) to indicate the exact position of the object at each frame .
	Directional guide	Directional guide	Arrows showing which direction an object is facing or moving (NOT the full path trace).
		Motion strength	Use of varying line weight (tapering or thickening strokes) or length to represent intensity, speed .
Style	Motion feeling	Motion feeling	Use of different style of the marks (e.g., different curveness) to indicate how the animation feel : e.g., it feels more bouncy/cartoonish/fun vs. it feels more rigid/realistic.
		Alignment arcs	Grid lines, horizontal/vertical guides, or ticks used to compare relative sizes or align elements across frames .
	Perspective guide	Perspective guide	Perspective grids, vanishing points, or receding lines used to indicate 3D depth and size reduction .
		Motion lines	Straight or arc strokes trailing behind a mover that indicate its path and suggest speed.
Motion amplifiers	Smears	Suppletion line	Parts of the figure are replaced by streaks, creating a blur that implies rapid motion .
		Polymorphism	Only a body part is repeated to show its movement through time.
	Impact	Backfixing line	Lines fixed to the background (not the mover), creating a background blur that implies the camera is following the subject.
		Circumfixing lines	Short ticks ringing a figure or body part to show vibration, wobble, or small repeated motion .
		Impact star	Starburst marks at the contact point to show the instant of collision .

Table 4: Notation Analysis Links with Labels.

ID	Link	Category	Sub-category	Label
1	https://pin.it/5QbXhFnrv	Geometric	force (animation principle)	Deformation (stretch/squash)
2	https://pin.it/3NdkokhLCa	Geometric	force (animation principle)	Deformation (stretch/squash)
3	https://pin.it/4GGkSeyqT	Geometric	force (animation principle)	Drag / Follow through
4	https://pin.it/7cEFXz5a1	Geometric	force (actual force)	Gravity / Weight Action vector
5	https://pin.it/7xT0kw3qB	Geometric	force (actual force) force (action-specific)	Gravity / Weight Action vector
6	https://pin.it/2KbUF3gOB	Geometric	force (action-specific)	Strike Force (e.g., punch, hit, kick) Rotational Force (e.g., swing / throw)
7	https://pin.it/1fVJMv8Cy	Geometric	force (actual force) force (action-specific)	Gravity / Weight Action vector
8	https://pin.it/1ZbFyGtU9	Geometric	force (animation principle) force (action-specific)	Action vector Overshoot
9	https://pin.it/7sSGiH0DR	Geometric	pose (living)	Line of action
10	https://pin.it/ApVBMmr2F	Geometric	pose (living)	Line of action
11	https://pin.it/35UVbjD2d	Geometric	pose (living)	Line of action Pose contour
12	https://pin.it/78flBm3xD	Geometric	pose (living)	Line of action
13	https://pin.it/1hnnaZolkv	Geometric	pose (living)	Line of action
14	https://pin.it/1SToAr9ap	Geometric	pose (living)	Line of action
15	https://pin.it/vVLQFZMLe	Geometric	pose (living)	Line of action
16	https://pin.it/U4m0Yk2Pl	Geometric	pose (living)	Line of action
17	https://pin.it/5EaiUN30B	Geometric	pose (living)	Line of action
18	https://pin.it/1bihRZfxY	Geometric	force (animation principle)	Overshoot Anticipation
19	https://pin.it/132HdUAtQ	Geometric	force (action-specific) pose (living)	Action vector Pose contour
20	https://pin.it/HVJXsewMU	Geometric	force (action-specific) pose (living)	Rotational Force (e.g., swing / throw) Pose contour Line of action
21	https://pin.it/5B5iMyWn9	Geometric	force (action-specific) force (actual force) force (animation principle)	Wind / Air flow Action vector Drag / Follow through
22	https://pin.it/3S9T5doki	Geometric	force (animation principle)	Deformation (twist/tilt/bend) Deformation (stretch/squash)
23	https://pin.it/6ru01bCf6	Geometric	style	Curved motion marks
24	https://pin.it/7wXYQDvrS	Geometric	pose (living)	Line of action
25	https://pin.it/O59kuS7xk	Geometric	pose (living)	Pose contour
26	https://pin.it/2vRm1Ewbh	Geometric	pose (non-living)	Elemental form
27	https://pin.it/7yogYNgSw	Geometric	pose (non-living)	Fabric folds
28	https://pin.it/cbz5ibJkR	Geometric	pose (non-living)	Elemental form
29	https://pin.it/79iafwYmS	Geometric	pose (non-living)	Elemental form
30	https://pin.it/7fo2nztk0	Geometric	pose (living) force (animation principle)	Deformation (twist/tilt/bend) Line of action
31	https://pin.it/7ziPTLQEY	Geometric	force (actual force) pose (living)	Gravity / Weight Pose contour
32	https://pin.it/4zfu3lQ0v	Geometric	pose (living)	Line of action Volumetric primitives
33	https://pin.it/4cVhHojkf	Geometric	pose (living) force (action-specific)	Line of action Action vector
34	https://pin.it/1GWIGRShf	Geometric	pose (living)	Line of action
35	https://pin.it/4HKx9LCB2	Geometric	pose (living) force (animation principle)	Pose contour Deformation (stretch/squash)

Continue on the next page....

Table 5: Notation Analysis Links with Labels (continued).

ID	Link	Category	Sub-category	Label
36	https://pin.it/1tNanb2J5	Geometric	path	Trajectory with spacing ticks
37	https://pin.it/2AMkUcm77	Geometric	path	Directional guide
38	https://pin.it/3QH8DCVJC	Geometric	path pose (non-living)	Directional guide Elemental form
39	https://pin.it/13h9TxWGl	Geometric	pose (living)	Line of action
40	https://pin.it/bm33gOPn7	Geometric	pose (living)	Line of action Pose contour
41	https://pin.it/3KBbe9lkf	Geometric	pose (living)	Volumetric primitives
42	https://pin.it/3Eqad1cx	Geometric	pose (living)	Line of action
43	https://pin.it/1XbSboKH9	Geometric	pose (living) force (action-specific)	Line of action Rotational Force (e.g., swing / throw)
44	https://pin.it/1RV910dNn	Geometric	force (actual force)	Deformation (twist/tilt/bend)
45	https://pin.it/OurMFYWr6	Geometric	force (action-specific)	Action vector
46	https://pin.it/3RkEmRbUf	Geometric	force (action-specific) force (animation principle)	Strike Force (e.g., punch, hit, kick) Anticipation
47	https://pin.it/4Q6ioCBOA	Geometric	pose (living)	Line of action Volumetric primitives
48	https://pin.it/1bEv0WrLq	Geometric	pose (non-living)	Fabric folds
49	https://pin.it/K7Zi2K12t	Geometric	force (animation principle) path	Deformation (stretch/squash) Trajectory
50	https://pin.it/4LAJF2OTc	Geometric	path	Trajectory
51	https://pin.it/2PXqLBUEK	Geometric	force (action-specific) force (actual force)	Deformation (twist/tilt/bend) Rotational Force (e.g., swing / throw)
52	https://pin.it/4spjPxyPR	Geometric	pose (living)	Line of action
53	https://pin.it/1pm1yARNr	Geometric	force (action-specific)	Rotational Force (e.g., swing / throw) Strike Force (e.g., punch, hit, kick)
54	https://pin.it/3V1Vx7OkS	Geometric	pose (living)	Line of action
55	https://pin.it/1YCJBWguK	Geometric	pose (living) force (action-specific)	Line of action Action vector Strike Force (e.g., punch, hit, kick)
56	https://pin.it/2EbnyUmPD	Geometric	pose (living)	Line of action
57	https://pin.it/puVzdr7gd	Geometric	pose (living)	Line of action
58	https://pin.it/1HUvtwUQ6	Geometric	path	Directional guide
59	https://pin.it/3nDQGrt8A	Geometric	path	Directional guide
60	https://pin.it/7ITbqxkG8	Geometric	pose (living) force (animation principle)	Deformation (stretch/squash) Pose contour
61	https://pin.it/3CKDxV5Eg	Geometric	path	Trajectory Directional guide
62	https://pin.it/40YbnR2GI	Geometric	pose (living)	Pose contour
63	https://pin.it/4Yj1PVyv7	Geometric	pose (living)	Line of action
64	https://pin.it/22GqWHFMc	Geometric	pose (living)	Pose contour
65	https://pin.it/30jX4VYJ9	Geometric	force (animation principle) force (actual force)	Drag / Follow through Gravity / Weight Wind / Air flow
66	https://pin.it/3hdQn7oPT	Geometric	force (action-specific) pose (living) path	Rotational Force (e.g., swing / throw) Contact (Friction/Push off) Action vector Trajectory with spacing ticks
67	https://pin.it/1uhVsRKOc	Geometric	force (actual force) force (action-specific) pose (living)	Gravity / Weight Action vector Volumetric primitives
68	https://pin.it/2rmhc9hOy	Geometric	force (actual force) pose (living)	Strike Force (e.g., punch, hit, kick) Pose contour
69	https://pin.it/24LKnlyGt	Geometric	style	Curved motion marks
70	https://pin.it/6SrjzpA3E	Geometric	pose (living) force (action-specific)	Line of action Rotational Force (e.g., swing / throw)
				Continue on the next page....

Table 6: Notation Analysis Links with Labels (continued).

ID	Link	Category	Sub-category	Label
71	https://pin.it/5f1ZEUeGW	Geometric	pose (living)	Pose contour
72	https://pin.it/ZYY33KA9O	Geometric	pose (non-living)	Fabric folds
73	https://pin.it/FGKEx7ObN	Geometric	pose (non-living)	Fabric folds
74	https://pin.it/3Sa2g1a3e	Geometric	force (animation principle) path force (action-specific)	Deformation (stretch/squash) Action vector Directional guide
75	https://pin.it/5TBehyCwR	Geometric	force (actual force) path	Contact (Friction/Push off) Trajectory
76	https://pin.it/7aIC3dggL	Geometric	pose (non-living)	Fabric folds
77	https://pin.it/31TTy0le3	Geometric	pose (non-living)	Elemental form
78	https://pin.it/6q0ftCUNW	Geometric	force (animation principle) force (actual force)	Deformation (stretch/squash) Deformation (twist/tilt/bend) Wind / Air flow
79	https://pin.it/1FZB1o8Os	Geometric	force (action-specific) pose (living)	Action vector Pose contour
80	https://pin.it/27k1wWgQa	Geometric	force (action-specific)	Rotational Force (e.g., swing / throw) Strike Force (e.g., punch, hit, kick)
81	https://pin.it/5vIFPlyJM	Geometric	force (actual force) force (action-specific)	Wind / Air flow Action vector
82	https://pin.it/29wFA4jDK	Geometric	force (animation principle)	Deformation (stretch/squash)
83	https://pin.it/5LKNGN5bc	Geometric	pose (living)	Line of action
84	https://pin.it/5aotlp3ZC	Geometric	pose (living)	Pose contour
85	https://pin.it/5oYpm9kzZ	Geometric	force (actual force) pose (living)	Deformation (twist/tilt/bend) Pose contour
86	https://pin.it/3m0tYgjwz	Geometric	force (animation principle) force (action-specific)	Drag / Follow through Action vector
87	https://pin.it/452hndcKi	Geometric	path	Directional guide
88	https://pin.it/2rkAKgXmT	Geometric	force (animation principle) force (action-specific)	Drag / Follow through Action vector
89	https://pin.it/52ZhA5el9	Geometric	pose (living)	Volumetric primitives
90	https://pin.it/7uAmm5uu5	Geometric	force (animation principle) force (action-specific) force (actual force)	Deformation (stretch/squash) Gravity / Weight Action vector
91	https://pin.it/barxlr0Ig	Geometric	path force (animation principle)	Deformation (stretch/squash) Trajectory
92	https://pin.it/4rhjWTOTD	Geometric	path	Directional guide
93	https://pin.it/6vEKiGYQd	Geometric	pose (living) force (action-specific)	Action vector Line of action
94	https://pin.it/49AoEF9E3	Geometric	force (action-specific) pose (living)	Action vector Volumetric primitives Line of action
95	https://pin.it/65A5OmwQt	Geometric	force (action-specific) pose (living)	Line of action Action vector
96	https://pin.it/EflyVorZ8	Geometric	force (action-specific)	Action vector Rotational Force (e.g., swing / throw)
97	https://pin.it/3qHZQreXG	Geometric	path	Directional guide
98	https://pin.it/2zsVedJP	Geometric	path	Directional guide
99	https://pin.it/1M7OAlt1F	Geometric	path force (actual force)	Trajectory Contact (Friction/Push off)
100	https://pin.it/5T9MbSeLA	Geometric	path	Trajectory with spacing ticks
101	https://pin.it/1gyN7HyFm	Geometric	path	Trajectory
102	https://pin.it/7hLEmwODJ	Geometric	path	Trajectory
103	https://pin.it/6I0kmI6o4	Spatial	perspective guide	Size reduction guide
104	https://pin.it/1uGII9IP9	Spatial	perspective guide	Size reduction guide
105	https://pin.it/4JHb9YZ6L	Spatial	perspective guide	Size reduction guide

Continue on the next page....

Table 7: Notation Analysis Links with Labels (continued).

ID	Link	Category	Sub-category	Label
106	https://pin.it/6BcGF6zWu	Spatial	perspective guide	Size reduction guide
107	https://pin.it/6zyEQa6tI	Spatial	perspective guide	Size reduction guide
108	https://pin.it/4Q2bWwdH1	Spatial	perspective guide	Size reduction guide
109	https://pin.it/1yRNEyVi5	Spatial	Alignment arcs	Porportion alignment
110	https://pin.it/6LHTJu0j7	Spatial	Alignment arcs	Alignment arcs
111	https://pin.it/wfkjtApIa	Spatial	Alignment arcs	Alignment arcs
112	https://pin.it/bRpfpObjCR	Spatial	Alignment arcs	Alignment arcs
113	https://pin.it/345tQpw2U	Spatial	Alignment arcs	Alignment arcs
114	https://pin.it/2JvHErrWL	Spatial	Alignment arcs	Alignment arcs
115	https://pin.it/6wD13GnYT	Spatial	Alignment arcs	Alignment arcs
116	https://pin.it/4mErVjNSB	Spatial	Alignment arcs	Alignment arcs
117	https://pin.it/7ipV05HHg	Spatial	Alignment arcs	Alignment arcs
118	https://pin.it/2AEpzmYXf	Spatial	Alignment arcs	Alignment arcs
119	https://pin.it/3obYNXpaD	Spatial	Alignment arcs	Alignment arcs
120	https://pin.it/4dZzRFntn	Spatial	Alignment arcs	Alignment arcs
121	https://pin.it/3z3IEwa9S	Spatial	Alignment arcs	Alignment arcs
122	https://pin.it/4q7c0dJVp	Spatial	Alignment arcs	Alignment arcs
123	https://pin.it/493T8KgaW	Spatial	Alignment arcs	Alignment arcs
124	https://pin.it/bhsKH21R0	Spatial	Alignment arcs	Alignment arcs
125	https://pin.it/qpdmQr5Dh	Amplifier	impact	Impact star
126	https://pin.it/19S05gaFo	Amplifier	smears	Suppletion line
127	https://pin.it/4Fp0jGKv2	Amplifier	impact	Impact star
128	https://pin.it/LYBv2V0h4	Amplifier	smears	Polymorphism
129	https://pin.it/3XcFsLmGp	Amplifier	smears	Polymorphism
130	https://pin.it/4D6CnOVF8	Amplifier	impact	Impact star
131	https://pin.it/6G7SVJK7U	Amplifier	impact	Circumfixing lines
132	https://pin.it/6iLOuIISY	Amplifier	smears	Backfixing line
133	https://pin.it/1bOmi5kKs	Amplifier	smears	Motion lines
134	https://pin.it/TO5sCVqhv	Amplifier	smears	Backfixing line
135	https://pin.it/7lrjeTPoh	Amplifier	smears	Motion lines