VideoSketcher: Video Models Prior Enable Versatile Sequential Sketch Generation

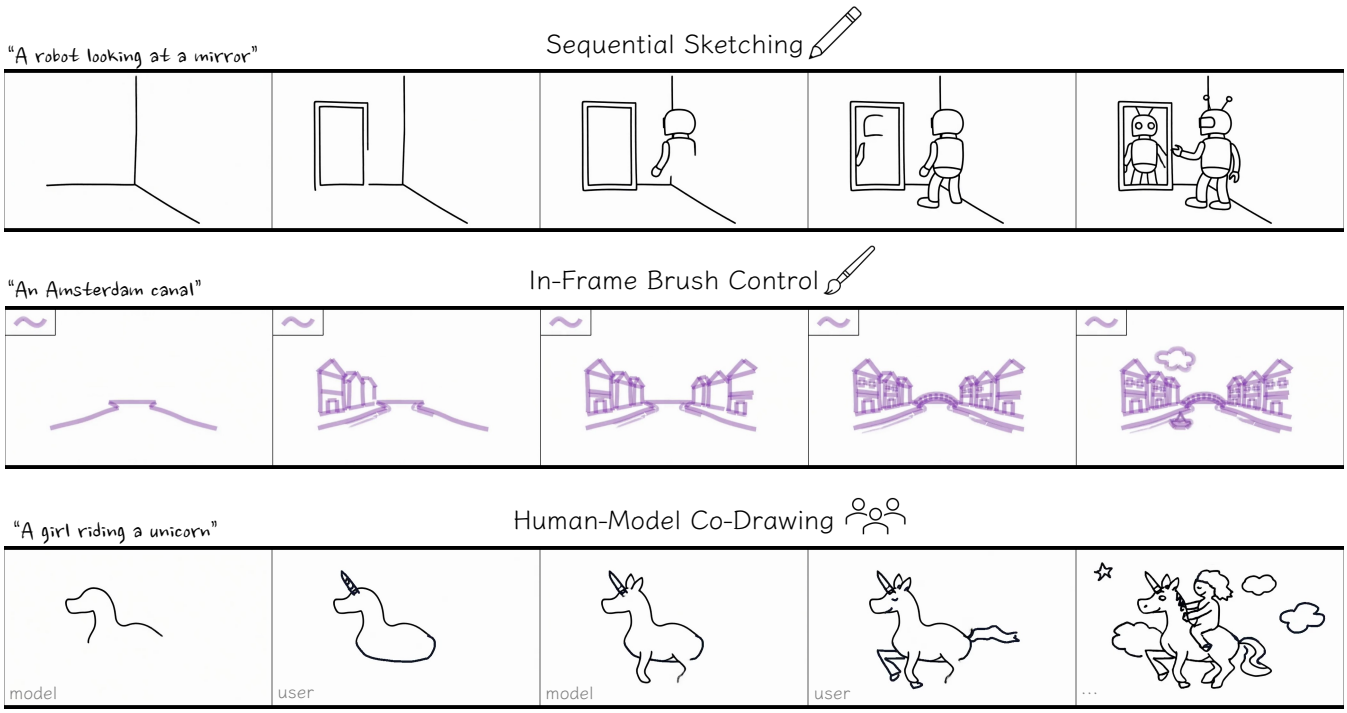

Fig. 1. VideoSketcher enables sequential sketch generation in pixel space via video diffusion priors. Given a text prompt, our method generates a step-by-step drawing process that follows natural stroke order with high visual quality (top). Our approach further supports user-specified brush styles (middle) and real-time human-model co-drawing through an autoregressive framework (bottom). Please see the full videos in our project page: https://videosketcher.github.io/

Sketching is inherently a sequential process, in which strokes are drawn in a meaningful order to explore and refine ideas. However, most generative models treat sketches as static images, overlooking the temporal structure that underlies creative drawing. We present a data-efficient approach for sequential sketch generation that adapts pretrained text-to-video diffusion models to generate sketching processes. Our key insight is that large language models and video diffusion models offer complementary strengths for this task: LLMs provide semantic planning and stroke ordering, while video diffusion models serve as strong renderers that produce high-quality, temporally coherent visuals. We leverage this by representing sketches as short videos in which strokes are progressively drawn on a blank canvas, guided by text-specified ordering instructions. We introduce a two-stage fine-tuning strategy that decouples the learning of stroke ordering from the learning of sketch appearance. Stroke ordering is learned using synthetic shape compositions with controlled temporal structure, while visual appearance is distilled from as few as seven manually authored sketching processes that capture both global drawing order and the continuous formation of individual strokes. Despite the extremely limited amount of human-drawn sketch data, our method generates high-quality sequential sketches that closely follow text-specified orderings while exhibiting rich visual detail. We further demonstrate the flexibility of our approach through extensions such as brush style conditioning and autoregressive sketch generation, enabling additional controllability and interactive, collaborative drawing.

1 Introduction

Sketches and drawings are a fundamental medium for exploring, communicating, and refining ideas [Fan et al. 2023; Tversky 2013]. Their expressive power lies not only in the final result, but also in the *process* of drawing itself: through the gradual accumulation of strokes, creators externalize thoughts, explore alternatives, and iteratively refine emerging concepts [Goldschmidt 1992; Tversky et al. 2003]. Computational models that generate sketches as sequential processes, rather than as static images, therefore have the potential to enable richer forms of human-machine interaction, including visual brainstorming, real-time feedback, and collaborative prototyping through a natural visual medium.

Yet modeling the *process* of drawing — rather than only its static final output — remains a significant challenge. The goal is not simply to generate strokes gradually, but to do so in a *meaningful, human-like order*, where structure is built through semantically coherent progressions rather than arbitrary stroke sequences. Prior approaches have taken important steps toward this goal, but face fundamental limitations. SketchRNN [Ha and Eck 2017] introduced autoregressive sketch generation trained directly on human drawing sequences, enabling the model to learn stroke ordering from data. However, this approach relies on millions of human-drawn

sketch sequences and is restricted to a fixed set of object categories with limited stylistic diversity.

More recently, SketchAgent [Vinker et al. 2025] demonstrated that multimodal large language models (LLMs) possess a surprising capacity for sequential sketch generation without requiring sketch-specific training. By prompting an LLM to output stroke coordinates as textual commands, SketchAgent generates drawings across a broad range of concepts, leveraging the model's semantic understanding to produce meaningful stroke orderings. However, this approach has an inherent bottleneck: while LLMs excel at deciding *what* to draw and in what *order*, they struggle with *how* to draw it. As a result, the generated sketches, although semantically coherent, tend to be overly simplistic and often lack visual quality.

In this work, we leverage text-to-video diffusion models for sequential sketch generation. Trained on large-scale video data, these models encode rich priors over visual appearance, motion, and temporal coherence. Our key insight is that video diffusion models and large language models offer *complementary* strengths for this task: LLMs provide semantic understanding that enables meaningful planning and stroke ordering, but are limited as visual renderers, whereas video diffusion models excel at high-quality visual synthesis but lack an intrinsic notion of drawing order. We combine these capabilities by using a video diffusion model as the "renderer", guided by an LLM that specifies what to draw and in which order.

We represent a sketch sequence as a short video in pixel space, in which black strokes are progressively drawn on a blank canvas. Despite the apparent gap between photorealistic video content and abstract hand-drawn sketches, we show that video diffusion models can be effectively distilled into sketch-like behavior using only a *handful* of carefully constructed examples.

A key challenge in this distillation process is teaching the model not only what sketches should look like, but also how they should unfold over time, following the ordering instructions specified by an LLM. Direct fine-tuning on hand-drawn sketches alone does not reliably yield such temporal control. To address this, we introduce a two-stage fine-tuning strategy. In the first stage, we train the model on a small, manually constructed dataset of basic geometric primitives, such as ovals, rectangles, triangles, and curves, arranged to exhibit fundamental compositional relationships inspired by Gestalt principles [Koffka 2013], including containment, adjacency, overlap, and grouping. Each composition is rendered using multiple drawing orders, teaching the model both a "visual vocabulary" of primitives and the ability to follow text-specified stroke sequences. In the second stage, we adapt the model to the visual style of hand-drawn sketches using as few as seven examples, transferring the learned temporal control to the target sketch domain.

Despite the extremely limited amount of real sketch data, our method generates high-quality sequential sketches across a diverse range of concepts, accurately following the ordering specified by the text prompt while exhibiting substantially richer visual detail and stronger temporal coherence (see Figure 1).

Beyond diffusion-based generation, we show that our distilled model can bootstrap autoregressive sketch generation by synthesizing additional training data, enabling interactive scenarios such as collaborative co-drawing. We also demonstrate that video diffusion models support brush style conditioning, allowing users to control stroke appearance using a simple visual cue. This enables brush-level control within a pixel-based generation framework — a capability usually associated with parametric stroke representations — and further illustrates the flexibility of video diffusion models for modeling drawing processes.

Together, these results suggest that pretrained video diffusion models offer a powerful and flexible prior for modeling drawing processes, providing a new perspective on sequential sketch generation that does not rely on large-scale sketch datasets or explicit parametric stroke representations.

2 Related Work

Sequential sketch generation. A common approach to sequential sketch generation represents sketches as explicit stroke sequences and trains models on large collections of human drawing data [Ha and Eck 2017; Tiwari et al. 2024; Wang et al. 2025; Xing et al. 2023a; Zhou et al. 2025]. Among these, SketchRNN [Ha and Eck 2017] pioneered this direction by introducing the QuickDraw dataset [Jonas et al. 2016], the largest collection of human-drawn sequential sketches. However, approaches that learn directly from such datasets are inherently constrained by the predefined categories and styles present in the training data. For QuickDraw, for example, this corresponds to at most 340 object categories with predominantly non-professional drawing quality.

To overcome this bottleneck, recent work has explored the use of large language models (LLMs) [Minaee et al. 2025; Zhao et al. 2025] to guide sequential visual generation. Because these models operate primarily over text, they are typically paired with external systems that translate language outputs into drawing actions or canvas edits [Hu et al. 2024; Shaham et al. 2024; W3C 1999; Wu et al. 2023; Yang et al. 2023]. SketchAgent [Vinker et al. 2025] frames sketch generation as a language-driven process, in which a multimodal LLM produces drawing instructions that are executed on a canvas. While this enables flexible, text-conditioned sequential sketching beyond fixed object categories, sketch quality remains constrained by a textual bottleneck: although LLMs excel at semantic reasoning and planning, they lack strong spatial and visual priors. As a result, the generated sketches tend to be overly simple and exhibit a child-like drawing style. In contrast, our method addresses this limitation by leveraging the rich visual priors of video models as powerful renderers of LLM commands, enabling substantially richer visual detail while preserving text-specified stroke ordering.

Recently, video models have been applied to painting reconstruction. *PaintsUndo* [Team 2024] and *PaintsAlter* [Zhang et al. 2025] aim to recover or reverse the creation process of an existing painting by predicting intermediate states conditioned on a completed image. While effective for painting reconstruction, these methods rely on $20k$ Procreate recordings for training and produce relatively coarse, frame-level progressions. In contrast, we pursue a fundamentally different goal: generating new sketches from text with explicit, stroke-by-stroke progression. This enables open-ended creative exploration in which the final result emerges through the drawing process rather than being predetermined. The slower, stroke-level progression naturally supports co-creation, allowing users to interpret and contribute as the sketch unfolds. Crucially, we demonstrate that this can be achieved using only a handful of training examples.

Finally, a related line of work uses reinforcement learning to train painting agents that produce sequential paintings for various tasks [Ganin et al. 2018; Mellor et al. 2019; Mihai and Hare 2021; Zhou et al. 2018]. However, these methods are not designed to model semantically meaningful stroke ordering and are typically restricted to narrow domains, such as faces or a small set of predefined objects.

VLM-guided vector sketch synthesis. Large pretrained diffusion and vision-language models (VLMs) provide strong semantic priors and have been widely used for static sketch and vector graphic generation [Podell et al. 2023; Radford et al. 2021; Rombach et al. 2022a; Saharia et al. 2022; Schuhmann et al. 2022]. A prominent line of work formulates sketch synthesis as an optimization problem over parametric vector strokes, which are iteratively refined under pixel-space guidance from pretrained models, typically large-scale VLMs, using differentiable rasterization [Arar et al. 2025; Choi et al. 2024; Gal et al. 2024; Jain et al. 2023; Li et al. 2020; Vinker et al. 2023, 2022; Xing et al. 2023b, 2024; Zhang et al. 2024].

While effective at producing semantically aligned static sketches, these approaches optimize all strokes jointly toward a final objective and do not explicitly model the temporal drawing process. As a result, they lack meaningful stroke ordering and are less suitable for interactive sketching scenarios.

Video priors and interactive video generation. Recent advances in video generation show that models trained on large-scale video data capture strong temporal structure and can serve as effective priors for new visual tasks [DeepMind 2025; HaCohen et al. 2024; Kong et al. 2025; OpenAI 2025; Wan et al. 2025; Wiedemer et al. 2025]. We build on this insight by adapting pretrained video generation models to learn sketching behavior in a few-shot setting.

Standard video diffusion models generate entire video clips jointly, making inference computationally expensive and limiting interactivity. Recent work, therefore, explores causal or autoregressive video models that use temporally causal attention to generate frames sequentially [Chen et al. 2024; Cui et al. 2025; Huang et al. 2025; Yang et al. 2025; Yin et al. 2024a,b, 2025a,b]. While these models may trade some visual fidelity for efficiency, they are better suited for human-in-the-loop applications. We adopt such an autoregressive model for sequential sketch generation, enabling collaborative sketching.

3 Preliminaries

Diffusion models generate samples by learning to reverse a gradual noising process. For high-dimensional data such as video, diffusion is typically performed in a compressed latent space obtained via a pretrained autoencoder, an approach commonly referred to as *latent diffusion* [Rombach et al. 2022b].

Given a video $V \in \mathbb{R}^{K\times H\times W\times 3}$ with K frames, a spatio-temporal variational autoencoder (VAE) encoder compresses it into a latent representation x_0. In this work, we build on Wan 2.1 [Wan et al. 2025], a pretrained open-source text-to-video diffusion model, which encodes the input video into a latent tensor $x_0 \in \mathbb{R}^{\frac{K}{4}\times\frac{H}{8}\times\frac{W}{8}\times 16}$. Both diffusion training and inference are performed in this latent space.

The diffusion model v_θ is a neural network—commonly implemented as a Diffusion Transformer (DiT) [Peebles and Xie 2023]—that learns to map samples from a Gaussian noise distribution $x_T \sim \mathcal{N}(0, I)$ to clean samples drawn from the data distribution $x_0 \sim p_{\text{data}}$ by progressively denoising x_T.

Recent video diffusion models adopt *rectified flow matching* [Lipman et al. 2023; Liu et al. 2022], which defines a linear interpolation path between clean data x_0 and Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$:

$$x_t = (1-t)\,x_0 + t\,\epsilon, \quad t \in [0,1], \tag{1}$$

where $t = 0$ corresponds to clean data and $t = 1$ corresponds to pure noise. The network v_θ is trained to predict the corresponding velocity field $v = \epsilon - x_0$ by minimizing the following objective:

$$\mathcal{L} = \mathbb{E}_{x_0,\epsilon,t}\left[\|v_\theta(x_t, t, y) - (\epsilon - x_0)\|^2\right], \tag{2}$$

where y denotes the text conditioning embedding.

At inference time, generation begins from noise $x_T \sim \mathcal{N}(0, I)$. Samples are obtained by integrating the differential equation

$$\frac{dx_t}{dt} = v_\theta(x_t, t, y) \tag{3}$$

from $t = 1$ to $t = 0$, yielding a clean latent representation x_0, which is then decoded into a video via the VAE decoder $\mathcal{D}(x_0)$.

4 Method

Our goal is to generate a video depicting a sequential sketching process conditioned on a text prompt, in which strokes follow the specified ordering and exhibit natural drawing behavior. We build on the strong priors of a pretrained text-to-video diffusion model [Wan et al. 2025] and adapt them to the sketch domain through a two-stage fine-tuning strategy. As we show, careful data construction is central to effective distillation: by disentangling stroke ordering from visual appearance across stages, the model learns complex sketching behavior from only a handful of examples.

4.1 Sketch Representation and Construction

We represent a sketch sequence as a short video $V \in \mathbb{R}^{K\times H\times W\times 3}$ in pixel space, depicting black strokes progressively drawn on a blank canvas. A natural sketching appearance requires modeling not only the global ordering of strokes, but also the local, continuous drawing within each stroke, mimicking the motion of a human hand across the canvas. To achieve this, we construct training data by capturing sketching processes as SVGs (Scalable Vector Graphics) [Cai et al. 2023] and rendering them via procedural stroke animation. An artist draws each sketch in Adobe Illustrator, which records both the stroke sequence and the drawing trajectory of each individual path. We then parse these SVG files and render them as videos by animating each stroke to appear gradually along its path, producing temporally structured sketching videos that preserve both stroke ordering and within-stroke progression (see Figure 2).

This representation offers several advantages. First, it enables fine-grained temporal control by ensuring that at most one stroke is introduced per frame, avoiding artifacts such as simultaneous stroke appearance. Second, it naturally accommodates different video model requirements, including resolution, frame count, and random seed, through simple adjustments to the rendering parameters. Third, as shown in Section 5.3, the parametric nature of SVGs enables the same stroke sequence to be rendered with different brush styles, providing additional training data for brush conditioning with no further manual effort.

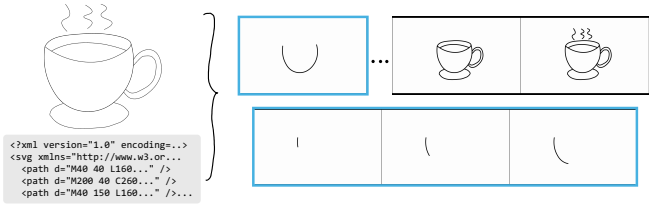

Fig. 2. **SVG-to-video sketch representation.** SVG paths are parsed and rendered sequentially into a video in which strokes are progressively drawn on a canvas. Each frame introduces at most one new stroke, ensuring clean temporal structure while enabling pixel-based video diffusion modeling.

Although SVGs offer a compact and structured representation, directly generating parametric sketches typically requires specialized architectures or inference-time optimization, often relying on large datasets and without explicitly modeling stroke order. By operating in pixel space, we instead leverage the full generative capacity of pretrained video diffusion models while still producing clean, parametric-like visual quality through sequential generation.

4.2 Text Prompt Construction

The text prompt must specify not only what to draw, but also the order in which elements should be drawn. We adopt a prompt format consisting of a brief description of the subject followed by an explicit, numbered sequence of drawing steps. For example:

Step-by-step sketch process of a desk lamp, following this drawing order:
1. Lampshade – a cone-shaped top part that directs the light downward.
2. Light bulb - …
…
7. Light beam emanating from the bulb.

Each step describes a semantic component of the object rather than low-level geometric primitives, and the ordering reflects natural drawing logic, such as sketching main forms before details and structural elements before secondary details.

At inference time, we use an LLM [OpenAI 2026] to generate structured drawing plans from high-level user input. This design exploits the LLM's strengths in semantic decomposition and planning [Vinker et al. 2025], while delegating visual realization and temporal rendering to the video diffusion model.

4.3 Two-Stage Finetuning

While video diffusion models encode strong visual priors, they lack an intrinsic notion of meaningful drawing order. As we demonstrate, naive fine-tuning on sketch videos can produce sketch-like appearance but often yields arbitrary or inconsistent stroke sequences. The central challenge, therefore, is to achieve both high visual fidelity and explicit temporal control over the sketching process.

We address this with a two-stage fine-tuning strategy that disentangles these objectives. Before training on real-world sketches, we first teach the model a basic drawing "grammar": simple shapes, their spatial relationships, and how to follow ordering instructions. This design is inspired by how people learn to draw, starting with simple shapes and compositional rules before progressing to more complex subjects [Coss and Kellogg 1969; Edwards 1989].

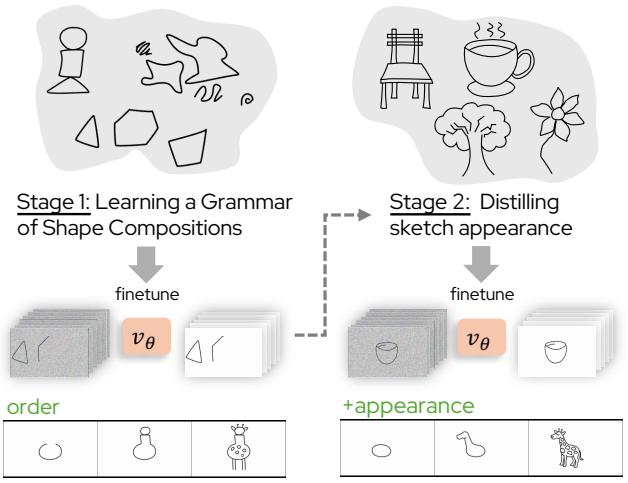

Fig. 3. **Two-stage fine-tuning scheme.** Left: synthetic sketches composed of simple geometric primitives teach drawing "grammar" and stroke ordering, independently of appearance. Right: a small set of human-drawn sketches of real-world objects adapts the model to a target visual style.

In both stages, fine-tuning uses standard video diffusion training with rectified flow matching. We do not modify the diffusion architecture or introduce task-specific losses; instead, learning is driven entirely by carefully constructed data, allowing the pretrained model to acquire sketch-specific behavior through data alone.

Learning the "Grammar" of Shape Compositions. We construct a small dataset of simple geometric primitives, including circles, ellipses, triangles, rectangles, polygons, curves, and lines, encoded as SVGs and rendered into sketch videos following the representation described in the previous section. These primitives are arranged in diverse spatial configurations inspired by Gestalt principles, such as containment (a circle inside a rectangle), adjacency (shapes placed side by side), overlap (partially occluding forms), and grouping (clustered elements). These relationships reflect the compositional building blocks underlying more complex sketches (see Figure 3). For each configuration, we vary the order in which the shapes are drawn, producing three distinct temporal variations per sketch.

Because the shapes are visually simple and semantically neutral, this dataset minimizes appearance-related variability and encourages the model to focus on learning temporal stroke ordering rather than object-specific visual details. Text prompts explicitly describe the intended drawing order, enabling the model to learn a direct correspondence between linguistic ordering cues, such as "first draw …, then draw …," and the temporal sequence of stroke additions. This synthetic pretraining stage is critical: as we demonstrate in Section 5.5, incorporating this initial fine-tuning stage with synthetic shape compositions significantly improves ordering fidelity.

Distilling Sketch Appearance. While training on primitive shape compositions is effective for learning stroke ordering, models trained only on such data tend to compose drawings directly from these primitives, resulting in sketches that lack the desired visual appearance. To bridge this gap, we perform a second fine-tuning stage

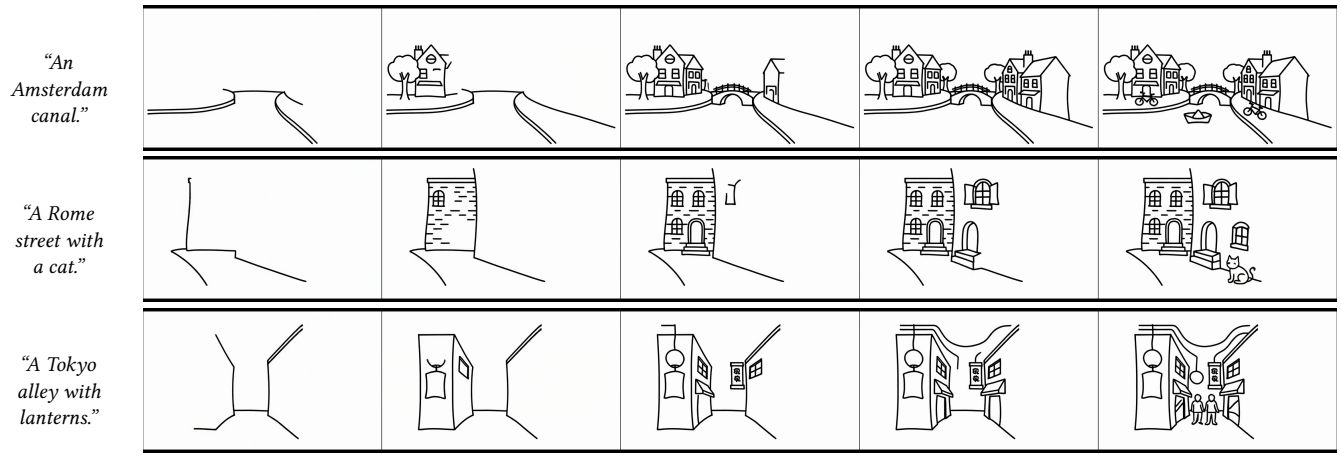

Fig. 4. **Qualitative results.** Results generated with our fine-tuned video model. Full video results are provided in the supplementary materials.

using a small set of seven real-world sketches drawn by an artist: a lamp, car, chair, tree, cup, butterfly, and flower (see Figure 3).

This stage adapts the model to the target visual aesthetic and level of abstraction that we aim to produce at inference time. Because the model has already learned to follow explicit ordering instructions during the synthetic pretraining stage, this fine-tuning primarily transfers appearance information rather than requiring the model to infer stroke ordering from a limited number of examples.

4.4 Brush Conditioning

We further demonstrate the flexibility of video models as priors for sequential sketch generation through a brush conditioning application. The goal is to allow users to control sketch style—both brush type and color—via a simple visual cue. Specifically, we provide the model with a small brush exemplar placed in the top-left corner of the canvas (see Figure 1), which guides stroke appearance throughout the sketching process without requiring an explicit parametric brush representation.

Training data is constructed using the same SVG sketches described earlier, rendering each sketch with multiple brush styles and colors, using six distinct brushes and eight colors in total. We then fine-tune an image-to-video diffusion model [Wan et al. 2025] using the same training procedure, with the first frame provided as a conditioning input image. In all cases, the first frame consists of a blank canvas augmented with the brush sample in the top-left corner, allowing the model to infer the desired brush appearance directly from the visual prompt.

4.5 Autoregressive Modeling

Our primary approach relies on diffusion-based video models that generate the entire sketch sequence jointly rather than predicting frames sequentially. In contrast, an autoregressive formulation — where each frame is conditioned on previously generated frames — is particularly well-suited for interactive sketching. Because sketching is inherently sequential, with each stroke building on the current canvas, this paradigm naturally aligns with drawing-based, user-in-the-loop interaction.

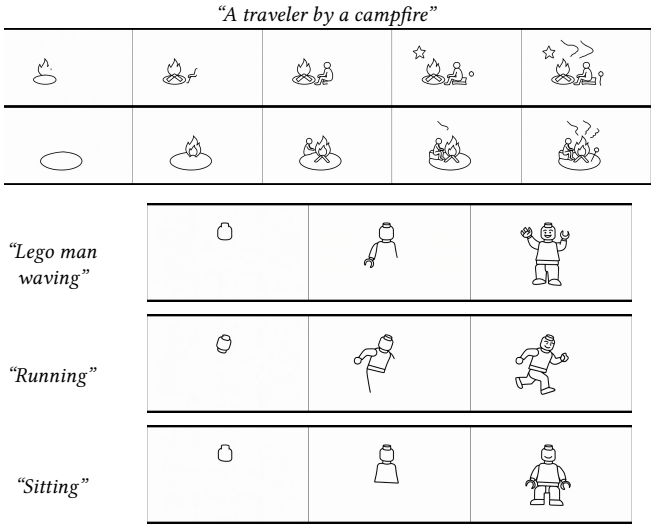

Fig. 5. **Diversity of sequential sketch generation.** Top: two seeds for the same prompt; Bottom: the same object in different settings/actions.

While autoregressive video models are less mature than diffusion-based approaches, recent work has demonstrated promising progress [Chen et al. 2024; Cui et al. 2025; Huang et al. 2025; Yin et al. 2024a,b, 2025a]. As these models continue to improve, our framework remains directly applicable, which we demonstrate by adapting our approach to an autoregressive video model.

A key challenge in autoregressive modeling is the need for substantially larger, high-quality training datasets [Yin et al. 2025a]. To address this, we use our diffusion-based sketch model, trained on a small set of real-world sketches, to generate a larger synthetic sketch dataset. This synthetic data is then used to fine-tune an autoregressive model, enabling sequential sketch generation in the autoregressive setting.

Table 1. **CLIP-based sketch recognition.** Average Top-1 and Top-5 accuracy of a CLIP zero-shot classifier evaluated on the last frame of 100 sketches from 50 categories. Results are reported as mean ± std across categories.

Method	Top-1	Top-5
Naive Prompting (Wan 2.1)	0.92 ± 0.03	0.99 ± 0.01
PaintsUndo (FLUX 2) [Team 2024]	1.00 ± 0.00	1.00 ± 0.00
SketchAgent [Vinker et al. 2025]	0.48 ± 0.05	0.71 ± 0.05
Human (QuickDraw [Jonas et al. 2016])	0.52 ± 0.05	0.70 ± 0.05
Ours	0.82 ± 0.04	0.96 ± 0.02
Ours (AR)	0.45 ± 0.04	0.70 ± 0.03

5 Results

Our method generates diverse sequential sketches, ranging from single objects to complex multi-element scenes, as shown in Figures 1, 4 and 14. Despite being trained on only 7 real sketches, it generalizes to complex scenes, including streets, canals, and alleys with multiple buildings, characters, and vehicles, producing clean lines, coherent perspective, and semantically meaningful stroke ordering (e.g., large structures first, fine details last). Moreover, as shown in Figure 5, our approach supports diverse outputs, either by varying the initial noise for a fixed prompt or by modifying the specified action.

5.1 Text-Conditioned Sketch Generation

We quantitatively evaluate how well the generated sketches depict the intended category. Following the evaluation protocol of SketchAgent [2025], we randomly sample 50 categories from the QuickDraw dataset [Jonas et al. 2016], and generate two sketches per category using different random seeds, yielding 100 sketches in total. We compare our method against several baselines, including naive prompting of Wan2.1 [Wan et al. 2025], PaintsUndo [Team 2024], SketchAgent [Vinker et al. 2025], and human-drawn sketches from the QuickDraw dataset. Since PaintsUndo requires a final frame as input, we provide final frames generated by FLUX2 [Labs 2025].

To quantify how well generated sketches depict the intended category, we follow standard practice [2023; 2022; 2025] and employ a CLIP ViT-B/32 zero-shot classifier on the final frame of each sequence. We report average Top-1 and Top-5 classification accuracy in Table 1. Our method achieves 82% Top-1 accuracy, substantially outperforming SketchAgent (48%) and human QuickDraw sketches (52%), while reaching 96% Top-5 accuracy compared to 71% and 70%, respectively. These results indicate that the fine-tuned video model generalizes effectively to a wide range of categories far beyond the seven sketches used during training.

We note that PaintsUndo attains high recognition scores because it is conditioned on a final image and reconstructs intermediate states that closely resemble the target, rather than generating sketches from text alone. As such, its performance is not directly comparable to methods that synthesize sketches from scratch. Wan 2.1 also achieves high final-frame recognition but, as shown in Section 5.2, fails to produce meaningful stroke-by-stroke progression, yielding nearly identical frames throughout the sequence. In contrast, our method produces both recognizable final sketches and coherent drawing processes, as shown in Figures 6 and 16.

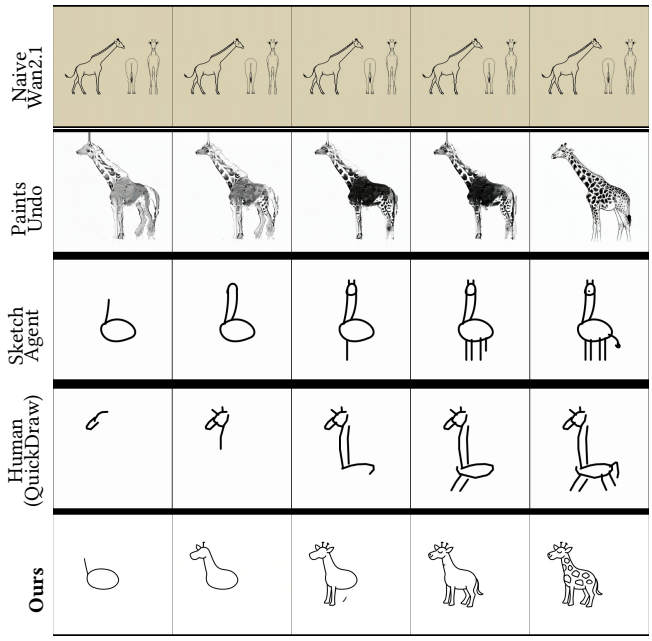

Fig. 6. **Qualitative comparison** of sketch generation across methods. The concept depicted is "a giraffe".

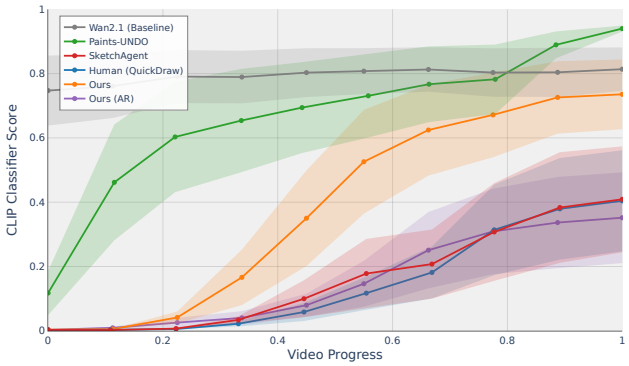

Fig. 7. **CLIP-based recognition over the sketching process.** Our method gradually increases semantic recognizability as the sketch progresses, in contrast to baselines that either collapse temporal progression or attain lower recognition. Shaded areas indicate variance across samples.

5.2 Sequential Sketching

Beyond semantic alignment with the text prompt, we evaluate the quality of the sketching process itself. As shown in Figure 15, our method produces clean, stroke-by-stroke progressions in which strokes are added incrementally according to the specified order. Varying the text instructions for the same concept results in distinct sketching trajectories, demonstrating control over stroke ordering.

To assess whether the temporal progression is semantically meaningful, we measure CLIP-based category recognition as a function of video progress, as shown in Figure 7. Representative frames are visualized in Figures 6 and 16 and in the supplementary material.

A naive prompting of Wan2.1 (gray) produces near-identical frames throughout the sequence, resulting in a flat recognition

Fig. 8. **Brush style control via visual prompting.** Left: brushes and colors seen during training. Right: generalization to unseen brushes and colors.

curve. PaintsUndo (green) saturates rapidly, reaching high recognition early in the video, which reflects its undo-based formulation where detailed structure appears upfront. While effective for its intended task. In addition, PaintsUndo produces detailed, painting-like outputs rather than the clean, vector-like sketches we target. SketchAgent (red) more closely follows human-like progression, but its outputs are often overly simplistic and sometimes fail to convey recognizable concepts (e.g., the cow example in Figure 16).

Our method closely tracks human progression while achieving substantially higher final accuracy (82% vs. 52% Top-1). As shown in Figure 6, sketches evolve through semantically meaningful stages that follow human-like ordering, such as drawing the body before the neck and head, while producing more detailed final appearances.

5.3 Brush Style Control

Beyond stroke ordering, our video-based formulation offers flexibility in visual style. As shown in Figures 8 and 13, conditioning on a brush exemplar in the first frame allows the model to reproduce the target brush's color and texture throughout the sketch. Notably, this generalizes to brush styles and colors not seen during training.

To quantitatively evaluate generalization to unseen brush styles, we apply five unseen colors and five unseen brushes to sketches from 30 object categories, using two random seeds per category, for a total of 1,500 samples. To measure alignment between the user-provided brush exemplar and the generated strokes, following prior style transfer work [Alaluf et al. 2024; Deng et al. 2024; Gatys et al. 2015], we compute the average ℓ_2 distance between Gram matrices extracted from three VGG-19 feature maps over stroke regions. As a baseline, we compare each output against all 25 brush exemplars used during evaluation, approximating the expected similarity to a randomly chosen brush. Our method achieves an average distance of 3.73 compared to 7.29 for the random baseline (a 49% reduction), indicating strong alignment with the target brush style.

5.4 Autoregressive Generation

Finally, we examine adapting our framework to autoregressive sketch generation, enabling interactive drawing scenarios that are difficult to support with diffusion-based models. As shown in Figure 11, the autoregressive model produces visually coherent sketches with clear stroke-by-stroke progression, although with slightly reduced visual fidelity compared to the diffusion-based approach.

We evaluate the autoregressive model using the same protocols as in Sections 5.1 and 5.2. For final-frame recognition (Table 1), it achieves 45% Top-1 and 70% Top-5 accuracy, comparable to human drawings (52%/70%) and SketchAgent (48%/71%). As shown in Figure 7, the recognition trajectory over time (purple) follows a gradual progression similar to human drawings (blue), indicating that the autoregressive formulation preserves meaningful temporal structure while enabling real-time interaction.

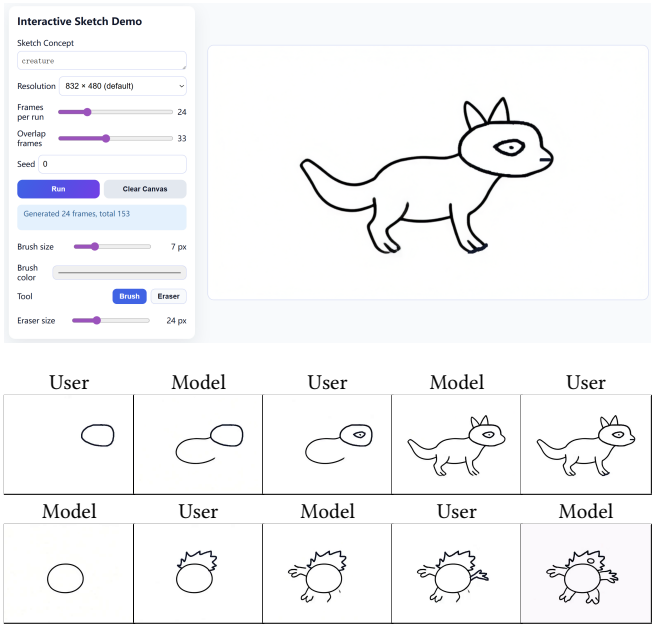

Fig. 9. **Co-Drawing.** Top: our interactive demo, where users draw alongside the model in real time. Bottom: turn-based co-drawing for "a creature."

To demonstrate interactivity, we built a prototype collaborative sketching interface in which the user and the model alternate adding strokes to a shared canvas. As shown in Figure 9, users can co-draw with the model in real time, with each adapting to the other's contributions to produce coherent sketches. This result demonstrates the feasibility of turn-based co-drawing and highlights the potential of autoregressive video models for interactive sketch generation.

5.5 Ablation Study

We ablate a key design choice in our training procedure, namely, separating the learning of stroke ordering from sketch appearance. We compare three variants: our full two-stage model, a model trained only on synthetic geometric primitives, and a model trained only on seven human-drawn sketches.

Qualitative results (Figure 10) show that the two-stage model is necessary to achieve both reliable ordering control and realistic sketch appearance. Training only on primitives yields more consistent ordering but produces primitive-like, less aesthetic, recognizable drawings. In contrast, training only on real sketches improves visual style, but often fails to follow the specified ordering. Combining both stages yields the best performance, transferring ordering fidelity learned from synthetic compositions into the target domain.

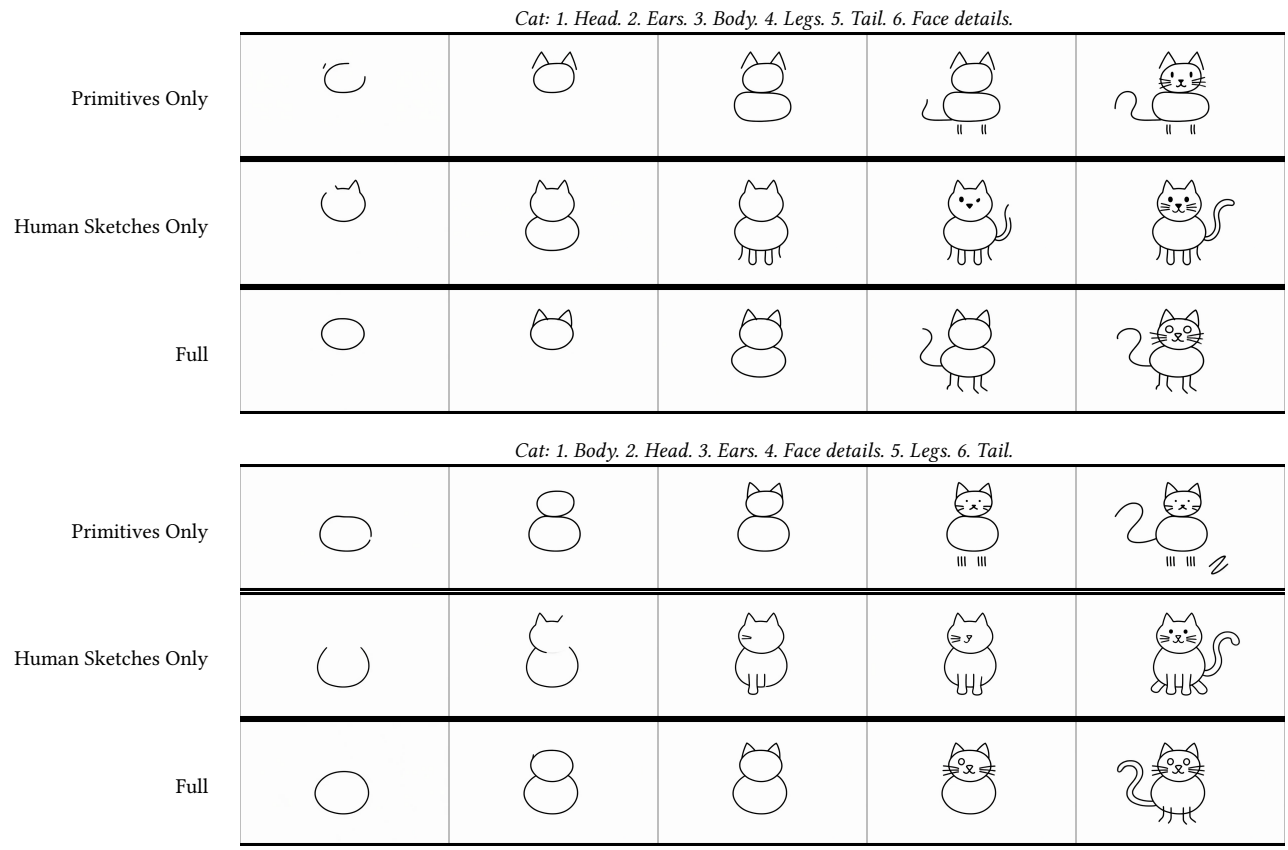

Fig. 10. **Ablation study qualitative comparison.** We compare sketching processes from models trained only on geometric primitives, only on real sketches, and the full two-stage model. The full model combines the strengths of both baselines, producing visually appealing sketches that follow the specified order.

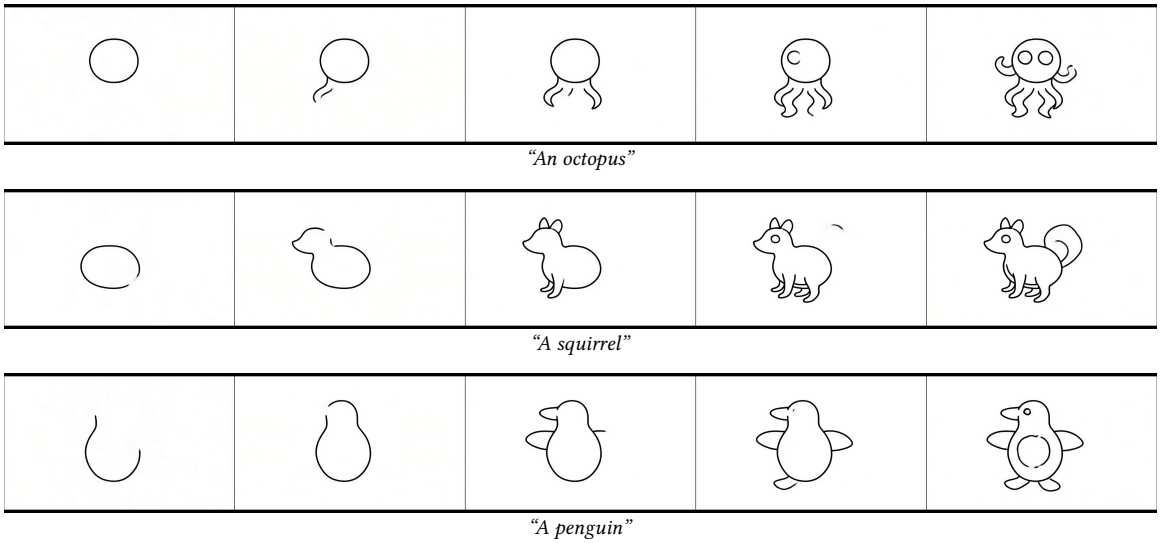

Fig. 11. **Autoregressive results.** Additional results from our autoregressive model, which enables interactive generation while maintaining visual quality comparable to diffusion-based results.

Table 2. **Ablation study.** Top: CLIP-based sketch recognition computed over our three model variants. Bottom: ordering fidelity, where an LLM compares target stroke orderings with those inferred from two generated sketches and selects the closer match or "Neither".

	Recognition		Ordering Fidelity		
Method / Comparison	Top-1	Top-5	Model A	Model B	Neither
Clip-Based Sketch Recognition					
Primitives Only	0.73	0.86	–	–	–
7 Human Sketches Only	0.88	0.96	–	–	–
Full Model (Ours)	**0.82**	**0.95**	–	–	–
Ordering Fidelity (LLM Preference)					
Primitives vs. 7-Human	–	–	**50.0**	37.2	12.8
Primitives vs. Full	–	–	26.9	**53.4**	19.7
7-Human vs. Full	–	–	29.6	**48.3**	22.1

To evaluate quantitatively, we measure CLIP Top-1 accuracy and ordering fidelity (Table 2). Ordering fidelity is assessed via LLM-based head-to-head comparisons against the target ordering, with details provided in the supplementary materials.

Quantitatively, the fully fine-tuned model achieves CLIP recognizability comparable to the model trained only on seven real sketches, indicating that visual aesthetic can be learned in a few-shot setting. The primitives-only model outperforms the real-sketch-only model in ordering fidelity, as expected given its focus on drawing grammar. When compared against both baselines, the fully fine-tuned model is strongly preferred in ordering fidelity, demonstrating that the two-stage training successfully combines ordering control with realistic sketch appearance. Although the full model outperforms the primitives-only model in ordering fidelity, this can be partly attributed to the simplicity of primitive-only sketches, which may not be recognizable enough for the LLM to reliably infer the drawn components and their order.

6 Limitations

While our method supports flexible sequential sketch generation across a wide range of prompts and styles, it has several limitations (see Figure 12). Operating in pixel space provides less explicit structural control than parametric stroke representations, which can occasionally lead to violations of sketching constraints, such as multiple strokes appearing within a single frame. Second, prompt adherence is not guaranteed. When the model has a strong visual prior, it may deviate from the instructions. For example, in the "tiger roaring" prompt, the model changes the action late in the video and introduces color. Third, performance depends on the video model's knowledge of the concept. Compared to LLMs, video models are less familiar with complex or specialized domains such as mathematics, which can lead to failures on highly unusual concepts even when detailed instructions are provided. Finally, while we demonstrate autoregressive sketch generation, the resulting outputs do not yet match the visual quality of the diffusion-based model, reflecting the present maturity of autoregressive video models.

7 Conclusions

We presented a data-efficient approach for sequential sketch generation that leverages pretrained text-to-video diffusion models as visual and temporal priors, guided by a large language model for semantic planning and stroke ordering. By representing sketches as videos and decoupling stroke ordering from sketch appearance, our method generates coherent, text-conditioned drawing processes with meaningful temporal structure, even when trained on only a handful of examples. We further show that these video priors support extensions such as brush style conditioning and autoregressive sketch generation, enabling greater controllability and interactive applications. Together, these results highlight the potential of pretrained video diffusion models as general-purpose priors for modeling structured, temporally grounded creative processes.

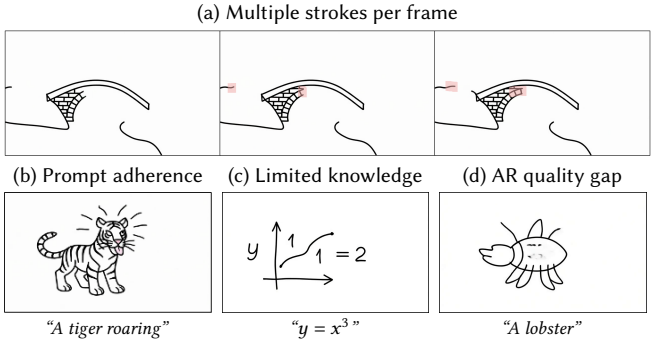

Fig. 12. **Limitations.** (a) Multiple strokes may appear together (in red). (b) Model's prior can override prompt. (c) Concepts outside the video model's knowledge are incorrectly depicted. (d) Reduced quality for AR outputs.

References

Yuval Alaluf, Daniel Garibi, Or Patashnik, Hadar Averbuch-Elor, and Daniel Cohen-Or. 2024. Cross-image attention for zero-shot appearance transfer. In *ACM SIGGRAPH 2024 conference papers*. 1–12.

Ellie Arar, Yarden Frenkel, Daniel Cohen-Or, Ariel Shamir, and Yael Vinker. 2025. Swiftsketch: A diffusion model for image-to-vector sketch generation. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*. 1–12.

Mu Cai, Zeyi Huang, Yuheng Li, Haohan Wang, and Yong Jae Lee. 2023. Delving into LLMs' visual understanding ability using SVG to bridge image and text. (2023).

Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. 2024. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *Advances in Neural Information Processing Systems* 37 (2024), 24081–24125.

Changwoon Choi, Jaeah Lee, Jaesik Park, and Young Min Kim. 2024. 3Doodle: Compact Abstraction of Objects with 3D Strokes. *ACM Trans. Graph.* 43, 4, Article 107 (July 2024), 13 pages. doi:10.1145/3658156

Richard Coss and Rhoda Kellogg. 1969. Analyzing Children's Art. *Leonardo* 4 (10 1969), 84. doi:10.2307/1572239

Justin Cui, Jie Wu, Ming Li, Tao Yang, Xiaojie Li, Rui Wang, Andrew Bai, Yuanhao Ban, and Cho-Jui Hsieh. 2025. Self-Forcing++: Towards Minute-Scale High-Quality Video Generation. *arXiv preprint arXiv:2510.02283* (2025).

Google DeepMind. 2025. Veo 3: High-fidelity, 8-second video generation with native audio. (2025). https://storage.googleapis.com/deepmind-media/veo/Veo-3-Tech-Report.pdf

Yingying Deng, Xiangyu He, Fan Tang, and Weiming Dong. 2024. Z-STAR+: A Zero-shot Style Transfer Method via Adjusting Style Distribution. arXiv:2411.19231 [cs.CV] https://arxiv.org/abs/2411.19231

Betty Edwards. 1989. *Drawing On the Right Side of the Brain: A Course in Enhancing Creativity and Artistic Confidence* (revised ed.). J. P. Tarcher.

Judith E Fan, Wilma A Bainbridge, Rebecca Chamberlain, and Jeffrey D Wammes. 2023. Drawing as a versatile cognitive tool. *Nature Reviews Psychology* 2, 9 (2023), 556–568.

Rinon Gal, Yael Vinker, Yuval Alaluf, Amit Bermano, Daniel Cohen-Or, Ariel Shamir, and Gal Chechik. 2024. Breathing Life Into Sketches Using Text-to-Video Priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 4325–4336.

Yaroslav Ganin, Tejas Kulkarni, Igor Babuschkin, SM Ali Eslami, and Oriol Vinyals. 2018. Synthesizing programs for images using reinforced adversarial learning. In *International Conference on Machine Learning*. PMLR, 1666–1675.

Leon A Gatys, Alexander S Ecker, and Matthias Bethge. 2015. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576* (2015).

Gabriela Goldschmidt. 1992. Serial sketching: visual problem solving in designing. *Cybernetics and System* 23, 2 (1992), 191–219.

David Ha and Douglas Eck. 2017. A Neural Representation of Sketch Drawings. *CoRR* abs/1704.03477 (2017). arXiv:1704.03477 http://arxiv.org/abs/1704.03477

Yoav HaCohen, Nisan Chiprut, Benny Brazowski, Daniel Shalem, Dudu Moshe, Eitan Richardson, Eran Levin, Guy Shiran, Nir Zabari, Ori Gordon, Poriya Panet, Sapir Weissbuch, Victor Kulikov, Yaki Bitterman, Zeev Melumian, and Ofir Bibi. 2024. LTX-Video: Realtime Video Latent Diffusion. arXiv:2501.00103 [cs.CV] https://arxiv.org/abs/2501.00103

Yushi Hu, Weijia Shi, Xingyu Fu, Dan Roth, Mari Ostendorf, Luke Zettlemoyer, Noah A Smith, and Ranjay Krishna. 2024. Visual sketchpad: Sketching as a visual chain of thought for multimodal language models. *Advances in Neural Information Processing Systems* 37 (2024), 139348–139379.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. 2025. Self Forcing: Bridging the Train-Test Gap in Autoregressive Video Diffusion. arXiv:2506.08009 [cs.CV] https://arxiv.org/abs/2506.08009

Ajay Jain, Amber Xie, and Pieter Abbeel. 2023. Vectorfusion: Text-to-svg by abstracting pixel-based diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1911–1920.

Jongejan Jonas, Rowley Henry, Kawashima Takashi, Kim Jongmin, and Fox-Gieg Nick. 2016. The Quick, Draw! - A.I. Experiment. https://github.com/googlecreativelab/quickdraw-dataset

Kurt Koffka. 2013. *Principles of Gestalt psychology*. routledge.

Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, Kathrina Wu, Qin Lin, Junkun Yuan, Yanxin Long, Aladdin Wang, Andong Wang, Changlin Li, Duojun Huang, Fang Yang, Hao Tan, Hongmei Wang, Jacob Song, Jiawang Bai, Jianbing Wu, Jinbao Xue, Joey Wang, Kai Wang, Mengyang Liu, Pengyu Li, Shuai Li, Weiyan Wang, Wenqing Yu, Xinchi Deng, Yang Li, Yi Chen, Yutao Cui, Yuanbo Peng, Zhentao Yu, Zhiyu He, Zhiyong Xu, Zixiang Zhou, Zunnan Xu, Yangyu Tao, Qinglin Lu, Songtao Liu, Dax Zhou, Hongfa Wang, Yong Yang, Di Wang, Yuhong Liu, Jie Jiang, and Caesar Zhong. 2025. HunyuanVideo: A Systematic Framework For Large Video Generative Models. arXiv:2412.03603 [cs.CV] https://arxiv.org/abs/2412.03603

Black Forest Labs. 2025. FLUX.2: Frontier Visual Intelligence. https://bfl.ai/blog/flux-2.

Tzu-Mao Li, Michal Lukáč, Gharbi Michaël, and Jonathan Ragan-Kelley. 2020. Differentiable Vector Graphics Rasterization for Editing and Learning. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)* 39, 6 (2020), 193:1–193:15.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. 2023. Flow Matching for Generative Modeling. arXiv:2210.02747 [cs.LG] https://arxiv.org/abs/2210.02747

Xingchao Liu, Chengyue Gong, and Qiang Liu. 2022. Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow. arXiv:2209.03003 [cs.LG] https://arxiv.org/abs/2209.03003

John FJ Mellor, Eunbyung Park, Yaroslav Ganin, Igor Babuschkin, Tejas Kulkarni, Dan Rosenbaum, Andy Ballard, Theophane Weber, Oriol Vinyals, and SM Eslami. 2019. Unsupervised doodling and painting with improved spiral. *arXiv preprint arXiv:1910.01007* (2019).

Daniela Mihai and Jonathon Hare. 2021. Learning to draw: Emergent communication through sketching. *Advances in Neural Information Processing Systems* 34 (2021), 7153–7166.

Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. 2025. Large Language Models: A Survey. arXiv:2402.06196 [cs.CL] https://arxiv.org/abs/2402.06196

OpenAI. 2025. Sora 2 System Card.

OpenAI. 2026. ChatGPT, GPT-5.2 Model. Large language model. https://chat.openai.com Accessed: January 19, 2026.

William Peebles and Saining Xie. 2023. Scalable Diffusion Models with Transformers. arXiv:2212.09748 [cs.CV] https://arxiv.org/abs/2212.09748

Dustin Podell, Zion English, Kyle Lacey, A. Blattmann, Tim Dockhorn, Jonas Muller, Joe Penna, and Robin Rombach. 2023. SDXL: Improving Latent Diffusion Models for High-Resolution Image Synthesis. *ArXiv* abs/2307.01952 (2023). https://api.semanticscholar.org/CorpusID:259341735

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. *CoRR* abs/2103.00020 (2021). arXiv:2103.00020 https://arxiv.org/abs/2103.00020

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022a. High-Resolution Image Synthesis with Latent Diffusion Models. arXiv:2112.10752 [cs.CV]

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022b. High-resolution image synthesis with latent diffusion models. 10684–10695 pages.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. 2022. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. arXiv:2205.11487 [cs.CV]

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. 2022. LAION-5B: An open large-scale dataset for training next generation image-text models. arXiv:2210.08402 [cs.CV]

Tamar Rott Shaham, Sarah Schwettmann, Franklin Wang, Achyuta Rajaram, Evan Hernandez, Jacob Andreas, and Antonio Torralba. 2024. A Multimodal Automated Interpretability Agent. In *Forty-first International Conference on Machine Learning*.

Paints-Undo Team. 2024. Paints-Undo GitHub Page. https://github.com/lllyasviel/Paints-UNDO

Adarsh Tiwari, Sanket Biswas, and Josep Lladós. 2024. SketchGPT: Autoregressive Modeling for Sketch Generation and Recognition. arXiv:2405.03099 [cs.CV] https://arxiv.org/abs/2405.03099

Barbara Tversky. 2013. Visualizing thought. In *Handbook of human centric visualization*. Springer, 3–40.

Barbara Tversky, Masaki Suwa, Maneesh Agrawala, Julie Heiser, Chris Stolte, Pat Hanrahan, Doantam Phan, Jeff Klingner, Marie-Paule Daniel, Paul Lee, et al. 2003. Sketches for design and design of sketches. *Human Behaviour in Design: Individuals, Teams, Tools* (2003), 79–86.

Yael Vinker, Yuval Alaluf, Daniel Cohen-Or, and Ariel Shamir. 2023. CLIPascene: Scene Sketching with Different Types and Levels of Abstraction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 4146–4156.

Yael Vinker, Ehsan Pajouheshgar, Jessica Y. Bo, Roman Christian Bachmann, Amit Haim Bermano, Daniel Cohen-Or, Amir Zamir, and Ariel Shamir. 2022. CLIPasso: Semantically-Aware Object Sketching. *ACM Trans. Graph.* 41, 4, Article 86 (2022). https://doi.org/10.1145/3528223.3530068

Yael Vinker, Tamar Rott Shaham, Kristine Zheng, Alex Zhao, Judith E Fan, and Antonio Torralba. 2025. SketchAgent: Language-Driven Sequential Sketch Generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*. 23355–23368.

W3C. 1999. *Scalable Vector Graphics (SVG)*. https://www.w3.org/Graphics/SVG

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu 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 [cs.CV] https://arxiv.org/abs/2503.20314

Jiawei Wang, Zhiming Cui, and Changjian Li. 2025. VQ-SGen: A Vector Quantized Stroke Representation for Creative Sketch Generation. arXiv:2411.16446 [cs.CV] https://arxiv.org/abs/2411.16446

Thaddäus Wiedemer, Yuxuan Li, Paul Vicol, Shixiang Shane Gu, Nick Matarese, Kevin Swersky, Been Kim, Priyank Jaini, and Robert Geirhos. 2025. Video models are zero-shot learners and reasoners. arXiv:2509.20328 [cs.LG] https://arxiv.org/abs/2509.20328

Chenfei Wu, Shengming Yin, Weizhen Qi, Xiaodong Wang, Zecheng Tang, and Nan Duan. 2023. Visual chatgpt: Talking, drawing and editing with visual foundation models. *arXiv preprint arXiv:2303.04671* (2023).

XiMing Xing, Chuang Wang, Haitao Zhou, Jing Zhang, Qian Yu, and Dong Xu. 2023a. DiffSketcher: Text Guided Vector Sketch Synthesis through Latent Diffusion Models. In *Thirty-seventh Conference on Neural Information Processing Systems*. https://openreview.net/forum?id=CY1xatvEQj

XiMing Xing, Chuang Wang, Haitao Zhou, Jing Zhang, Qian Yu, and Dong Xu. 2023b. DiffSketcher: Text Guided Vector Sketch Synthesis through Latent Diffusion Models. In *Advances in Neural Information Processing Systems*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (Eds.), Vol. 36. Curran Associates, Inc., 15869–15889. https://proceedings.neurips.cc/paper_files/paper/2023/file/333e67fc4728f147d31608db3ca78e09-Paper-Conference.pdf

Ximing Xing, Haitao Zhou, Chuang Wang, Jing Zhang, Dong Xu, and Qian Yu. 2024. SVGDreamer: Text Guided SVG Generation with Diffusion Model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 4546–4555.

Shuai Yang, Wei Huang, Ruihang Chu, Yicheng Xiao, Yuyang Zhao, Xianbang Wang, Muyang Li, Enze Xie, Yingcong Chen, Yao Lu, Song Han, and Yukang Chen. 2025. LongLive: Real-time Interactive Long Video Generation. arXiv:2509.22622 [cs.CV] https://arxiv.org/abs/2509.22622

Zhengyuan Yang, Jianfeng Wang, Linjie Li, Kevin Lin, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. 2023. Idea2img: Iterative self-refinement with gpt-4v (ision) for automatic image design and generation. *arXiv preprint arXiv:2310.08541* (2023).

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and Bill Freeman. 2024a. Improved distribution matching distillation for fast image synthesis. *Advances in neural information processing systems* 37 (2024), 47455–47487.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. 2024b. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 6613–6623.

Tianwei Yin, Qiang Zhang, Richard Zhang, William T Freeman, Fredo Durand, Eli Shechtman, and Xun Huang. 2025a. From Slow Bidirectional to Fast Autoregressive Video Diffusion Models. *CVPR*.

Tianwei Yin, Qiang Zhang, Richard Zhang, William T Freeman, Fredo Durand, Eli Shechtman, and Xun Huang. 2025b. From slow bidirectional to fast autoregressive video diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*. 22963–22974.

Lvmin Zhang, Chuan Yan, Yuwei Guo, Jinbo Xing, and Maneesh Agrawala. 2025. Generating Past and Future in Digital Painting Processes. *ACM Transactions on Graphics (SIGGRAPH 2025)* 44, 4, Article 127 (2025), 13 pages.

Peiying Zhang, Nanxuan Zhao, and Jing Liao. 2024. Text-to-Vector Generation with Neural Path Representation. *ACM Trans. Graph.* 43, 4, Article 36 (July 2024), 13 pages. doi:10.1145/3658204

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2025. A Survey of Large Language Models. arXiv:2303.18223 [cs.CL] https://arxiv.org/abs/2303.18223

Jin Zhou, Yi Zhou, Hongliang Yang, Pengfei Xu, and Hui Huang. 2025. StrokeFusion: Vector Sketch Generation via Joint Stroke-UDF Encoding and Latent Sequence Diffusion. arXiv:2503.23752 [cs.GR] https://arxiv.org/abs/2503.23752

Tao Zhou, Chen Fang, Zhaowen Wang, Jimei Yang, Byungmoon Kim, Zhili Chen, Jonathan Brandt, and Demetri Terzopoulos. 2018. Learning to sketch with deep q networks and demonstrated strokes. *arXiv preprint arXiv:1810.05977* (2018).

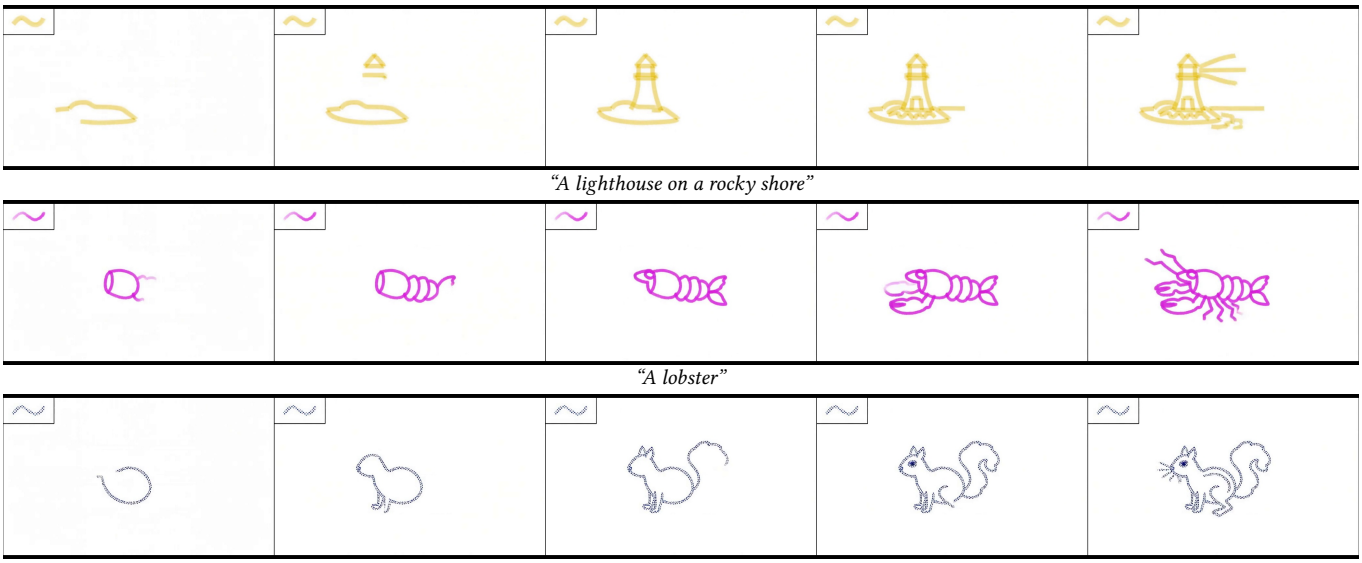

Fig. 13. **Additional results for brush style control.** We show concepts drawn with seen (left) and unseen (right) brush styles and colors.

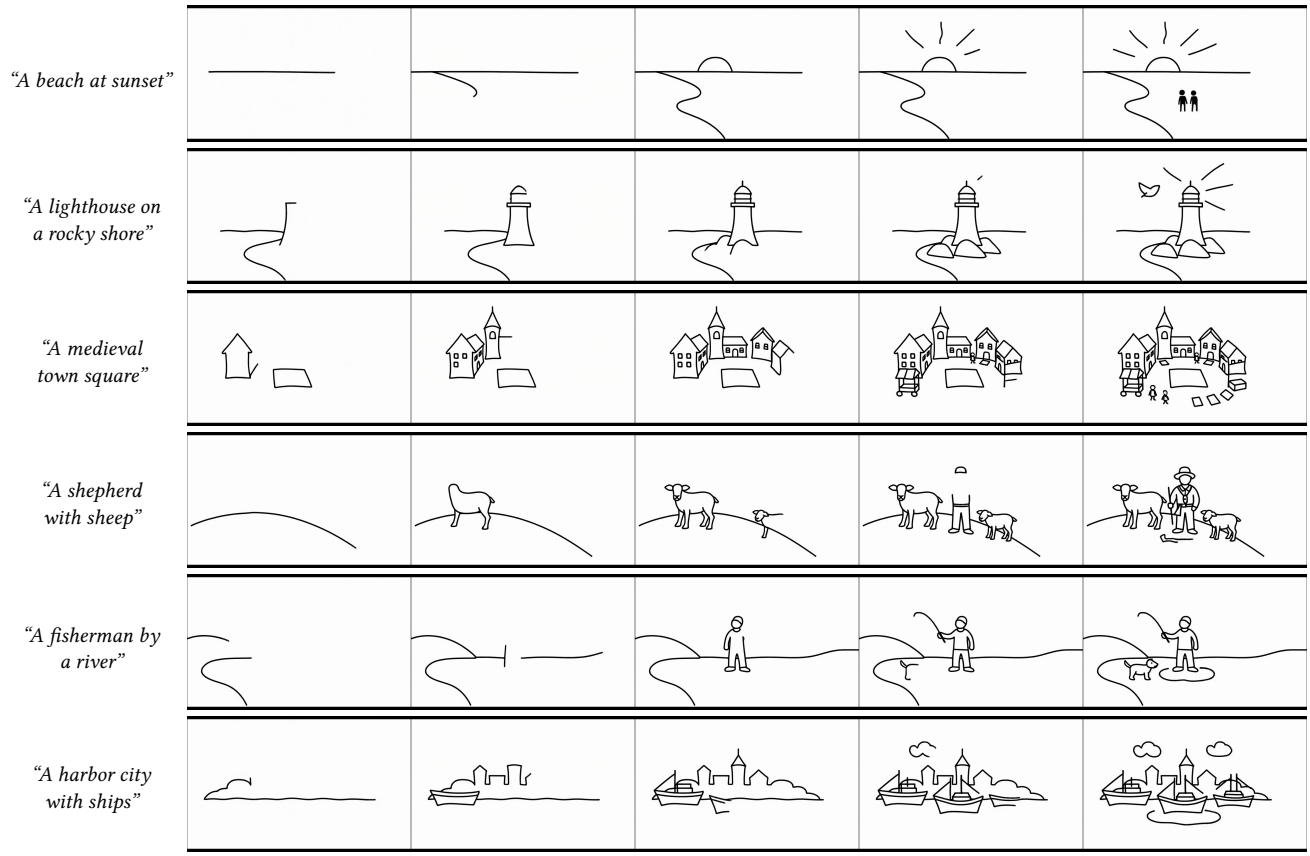

Fig. 14. **Qualitative results.** Additional results generated with our fine-tuned text-to-video model. Full video results are provided in the supplementary.

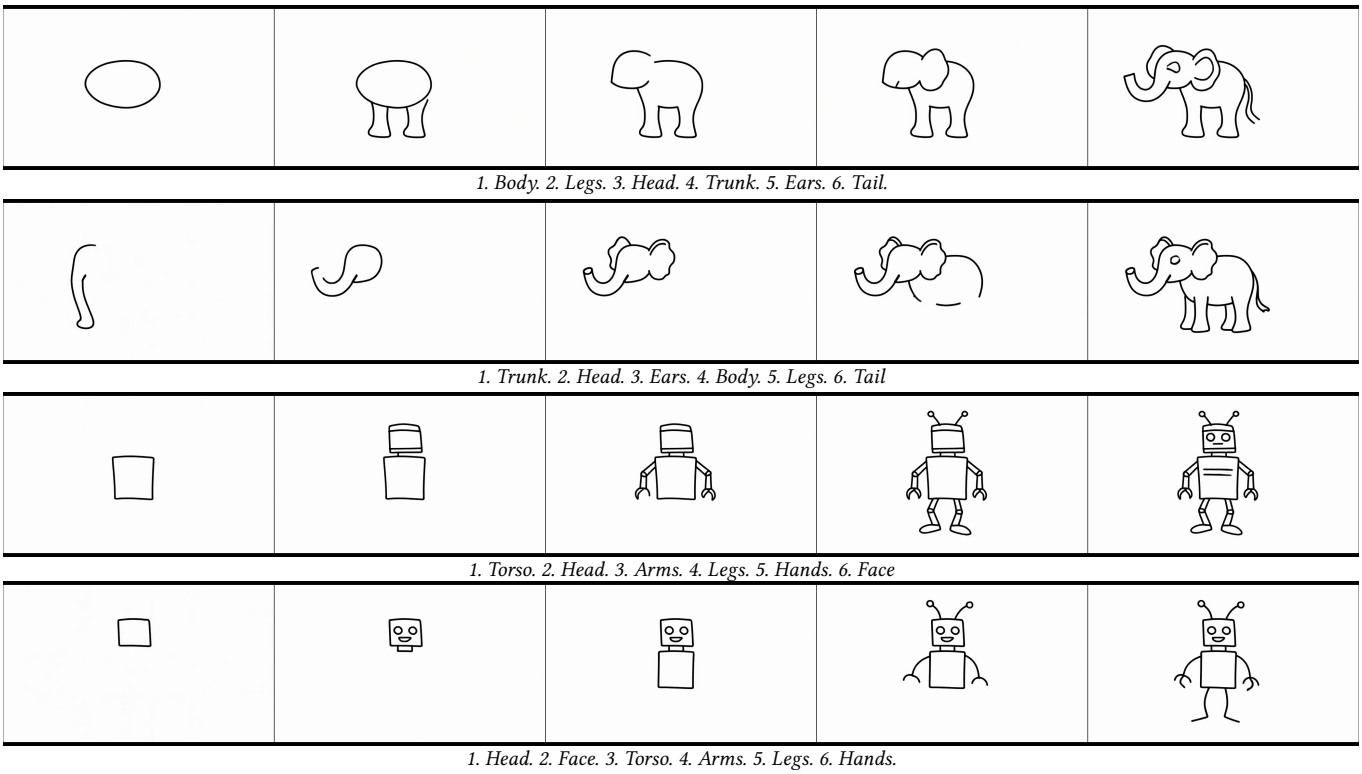

Fig. 15. **Text-specified stroke ordering.** Each row shows the same concept generated using a different text prompt that specifies a distinct drawing order.

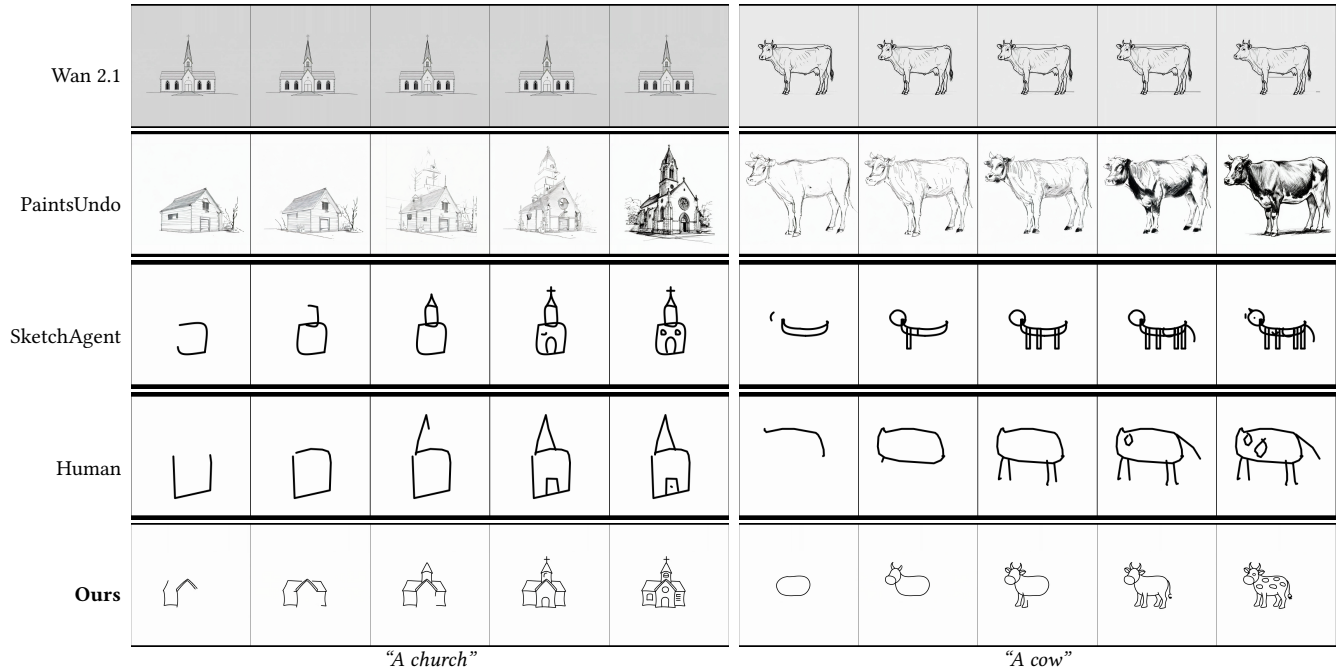

Fig. 16. **Additional qualitative comparison of sequential sketch generation across methods.** The Human drawings are taken from QuickDraw [Jonas et al. 2016]. Full video results are provided in the supplementary materials.

VideoSketcher: Leveraging Video Model Priors for Versatile Sequential Sketch Generation Supplementary Material

Fig. 17. Six brush styles (left) and 8 colors (right) used for training our brush-conditioned model.

A Implementation Details

We use the pretrained Wan 2.1 14B [Wan et al. 2025] diffusion model as our base video backbone. Fine-tuning is performed using LoRA adapters applied to the attention layers and the first two layers of the feed-forward networks in the diffusion transformer. This design enables efficient adaptation while mitigating overfitting in the low-data regime. All diffusion models are trained using a LoRA rank of 32 and the standard rectified flow matching loss.

Training is conducted on 7 NVIDIA A100 GPUs with a batch size of 1 and a learning rate of 1e−4. The synthetic shape training stage runs for 700 epochs on 15 videos. The additional fine-tuning on the 7 human-drawn sketches is then applied for 700 additional epochs. Across all experiments, both training and inference are performed at a resolution of 480×832 with 81 frames. Training overall takes about 22 hours. At inference, for the best performance, we apply the trained model with 50 inference steps (though using 10 steps also produces plausible results and can save inference time). Inference of a video with 81 frames with 50 denoising steps and resolution 480×832 takes 16 minutes on a single A100 GPU.

Brush-Conditioned Image-to-Video Model. For brush style conditioning, we use the image-to-video variant of the Wan 2.1 14B model. The training data is augmented with 6 brush styles and 8 colors (shown in Figure 17), resulting in 720 samples for the shape training stage and 336 samples for the human-drawn real sketches stage. The training process follows the same settings as described above for the text-to-video diffusion model.

Autoregressive Model. For autoregressive sketch generation, we use the CausVid [Yin et al. 2025a] model as the base video model, which is a fine-tuned autoregressive variant of Wan 2.1 1.3B (a smaller model compared to the 14B model, with slightly reduced quality). We construct the training set using 43 videos generated by our 14B text-to-video diffusion model, together with 7 real-world sketches, yielding a total of 50 training videos. The 43 videos were generated by randomly sampling categories from the QuickDraw dataset (ones not used in our evaluation setup) and generating detailed prompts for them. We train the full diffusion transformer for 2700 epochs with a learning rate of 2e−6, using a regression loss on the ODE trajectory. Qualitative results on QuickDraw prompts are shown in Figures 35 and 36. Inference of a video with 81 frames and resolution 480×832 takes about 11 seconds on a single A100 GPU. This means it takes about 4 seconds to produce 24 frames, which is our default step size in the interactive demo, enabling real-time interaction.

Quantitative Metrics. Stroke Ordering. In the ablation study presented in the main paper, we report quantitative measurements of ordering fidelity for three model variants: a model trained solely on simple geometric primitives, a model trained on only seven human-drawn sketches, and our full model trained using the proposed two-stage approach.

Since no established metrics exist for evaluating ordering fidelity with respect to a text prompt, we adopt an LLM-guided evaluation protocol consisting of two stages. First, given a video generated by a model, we prompt an LLM to extract the sequence in which semantic parts are drawn (e.g., "1. Body, 2. Head, 3. Face, …"). To ensure consistent terminology across models, we also provide the LLM with the target ordering, which constrains the vocabulary used to describe the parts (e.g., enforcing the term "Body" rather than alternatives such as "Torso").

Next, we perform pairwise, head-to-head comparisons between each combination of the three models. For each pair, the LLM compares the extracted orderings against the target ordering and selects the model that better adheres to it. When both models deviate from or match the target ordering to a similar extent, the LLM is allowed to return "Neither," resulting in a tie. This evaluation is conducted over 100 prompts from QuickDraw and three random seeds, with the averaged results reported in Table 2 of the main paper.

B Additional Experiments

B.1 Out-of-Distribution Concept Generation

In this section we evaluate our method on concepts requiring specialized knowledge. We follow the experimental setup of SketchAgent [Vinker et al. 2025], where three categories requiring general knowledge are defined: Scientific Concepts, Diagrams, and Notable Landmarks, with ChatGPT used to produce 10 random textual concepts per category. We extend this setup by adding a Functions category, as functions can be thought of as drawings requiring specialized knowledge while being easier to evaluate for correctness. In summary, we use the following categories and concepts:

- **Scientific Concepts**
  Double-slit experiment, Pendulum motion, Photosynthesis, DNA replication, Newton's laws of motion, Electromagnetic spectrum, Plate tectonics, Quantum entanglement, Cell division (mitosis), Black hole formation.
- **Diagrams**
  Circuit diagram, Flowchart, Organizational chart, ER diagram (Entity-Relationship), Venn diagram, Mind map, Gantt chart, Network topology diagram, Pie chart, Decision tree.
- **Notable Landmarks**
  Taj Mahal, Eiffel Tower, Great Wall of China, Pyramids of Giza, Statue of Liberty, Colosseum, Sydney Opera House, Big Ben, Mount Fuji, Machu Picchu.
- **Functions**
  $y = x^2 y = \sqrt{x}, y = x^3, y = log(x), y = e^x, y = \frac{1}{x}, y = sin(x), y = |x|, y = 2x, y = x.$

For each concept, we generate three random sketch sequences using our method and compare with SketchAgent and Wan2.1 (as a baseline, without any fine-tuning). Representative results showing the last frame of each produced video are presented in Figures 18 to 20 and 22.

The Wan2.1 results reveal what the video model already knows prior to fine-tuning. Concepts familiar to the base model will more naturally transfer to our fine-tuned model. This is evident in the Landmarks category (Figure 20), where our results are highly detailed and recognizable, reflecting the model's prior knowledge. For Scientific Concepts, the pattern is more nuanced: where the base model is limited (e.g., pendulum motion), our model inherits these limitations, while for concepts like Newton's laws and photosynthesis, our model performs well.

This experiment also highlights the complementary strengths of video-based and LLM-based approaches. For Landmarks (Figure 20), SketchAgent produces overly simplistic outputs with low visual quality (e.g., Eiffel Tower, Statue of Liberty), while ours are detailed and recognizable. However, for Scientific Concepts (Figure 18), the strong priors of LLMs enable SketchAgent to capture the correct structure and rules — achieving correctness despite lower visual aesthetics. This contrast is most apparent in the Functions category: both Wan2.1 and our method fail to draw functions properly, while SketchAgent's LLM backbone enables more precise results.

We additionally assess recognizability quantitatively via classification with GPT-4o under two settings: (1) *multi-choice*, where the model selects among the 10 category concepts or "none" (easier setting), and (2) *free-text*, where the model describes the sketch without provided options and we analyze whether the output matches the class (stricter setting). Results are reported in Figure 21. The bar chart reflects the patterns observed visually: there is clear correlation between base model success (red) and our model (green), where knowledge can be lost (as in Scientific Concepts), while SketchAgent struggles on Landmarks and concepts requiring high detail and visual quality.

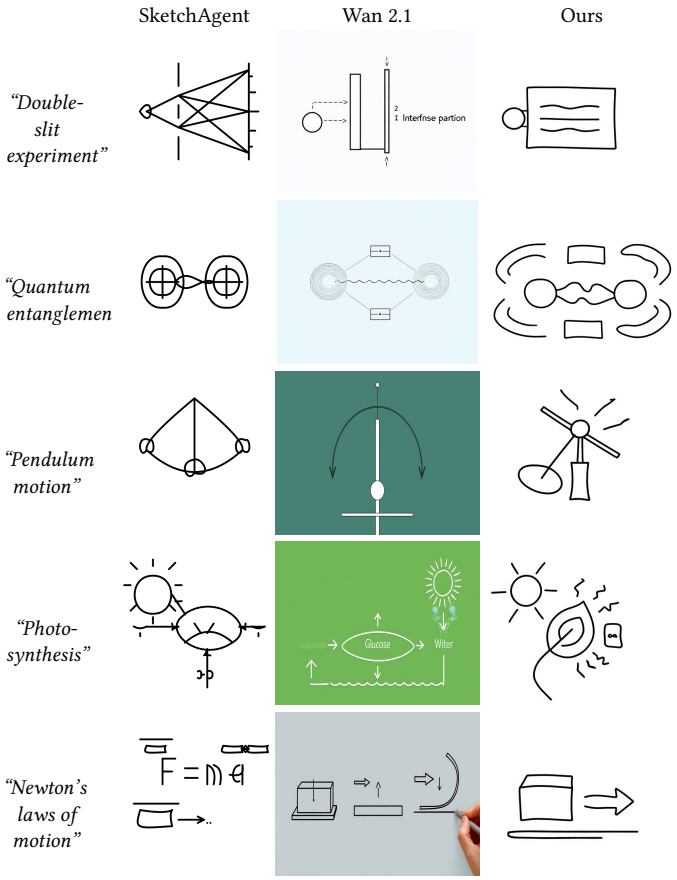

Fig. 18. **Scientific Concepts.** Representative results for scientific concepts across methods. SketchAgent leverages LLM knowledge to capture conceptual structure (e.g., the interference pattern in double-slit, F=ma in Newton's laws), though with limited visual detail. Wan 2.1 occasionally produces informative diagrams but often includes colored backgrounds and text labels rather than sketch-style outputs. Our method produces visually coherent sketches, inheriting both the strengths and limitations of the base model—performing well on concepts like photosynthesis while struggling with more abstract ones like pendulum motion.

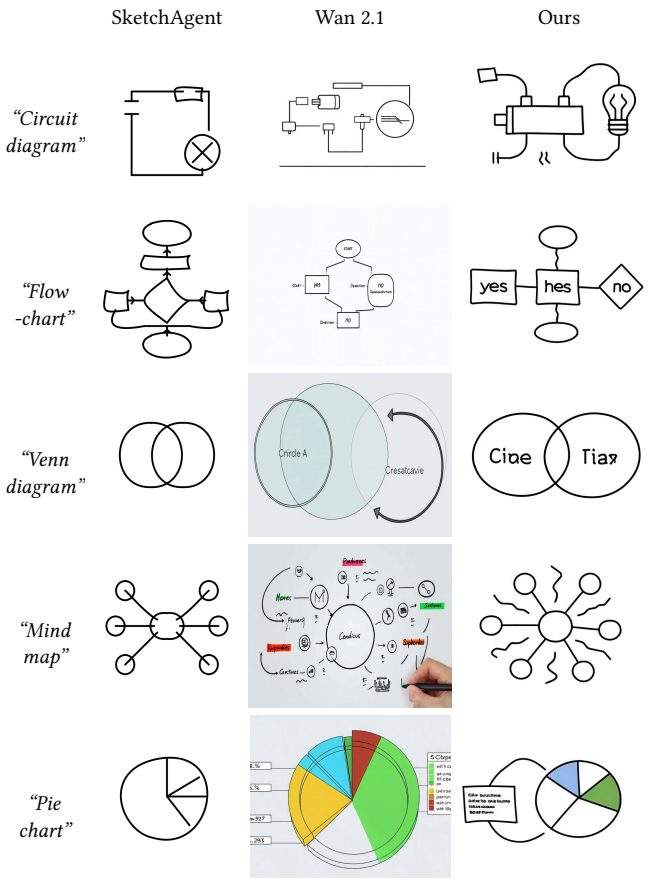

Fig. 19. **Diagrams.** Representative results for diagram concepts. SketchAgent produces structurally correct but visually minimal outputs. Wan 2.1 generates detailed diagrams with text and color, deviating from a sketch aesthetic. Our method captures the visual structure of diagrams (e.g., connected nodes in flowcharts and mind maps, overlapping circles in Venn diagrams) with a cleaner sketch style, though text elements are often garbled or nonsensical.

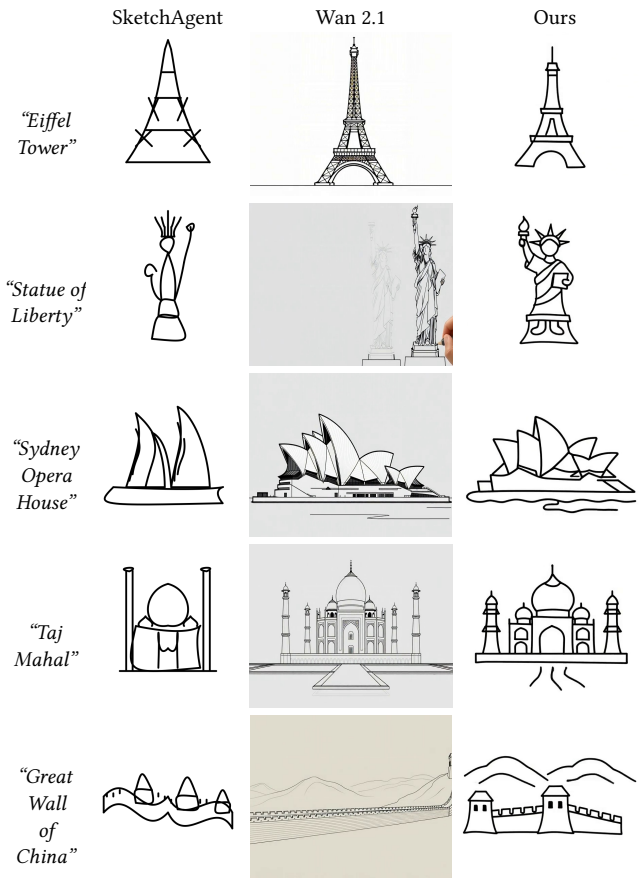

Fig. 20. **Notable Landmarks.** Representative results for landmark concepts. This category highlights the strength of video-based approaches: both Wan 2.1 and our method produce detailed, highly recognizable depictions, reflecting the strong prior knowledge of landmarks in video training data. In contrast, SketchAgent's outputs are overly simplistic and often fail to capture the iconic features (e.g., the Eiffel Tower reduced to basic triangles, the Statue of Liberty barely recognizable).

B.2 Quantitative Evaluation

B.3 Sketch Progression

To verify that generated sketches unfold sequentially rather than collapsing temporally, we measure the number of newly added pixels at each frame throughout the sketching process. Specifically, we compute the cumulative ratio of added pixels as a function of video progress, normalized so that the final frame equals 1 (see Figure 23). Both our diffusion-based and autoregressive models exhibit smooth, gradual accumulation curves, indicating that strokes are introduced incrementally across frames in a manner consistent with human drawing behavior. In contrast, baseline methods tend to introduce a large fraction of pixels early in the sequence, as also reflected in the qualitative results.

Table 3. **Stroke continuity evaluation.** We measure the frequency of frames containing multiple disjoint strokes on QuickDraw object concepts (typically simpler concepts) and scene-level prompts (typically more complex concepts).

Evaluation Set	Num. Samples	Total Frames	Multi-Stroke Frames	Ratio ↓
QuickDraw Concepts	200	15196	2975	19.58%
Scenes	80	6361	2351	36.96%

B.4 Multi-Stroke Emergence Evaluation

Because our models generate sketches directly in pixel space, they do not explicitly enforce stroke continuity. As a result, a single frame may contain multiple disjoint strokes that emerge simultaneously. We quantify this effect by computing the percent of frames that contain multiple strokes, reported in Table 3.

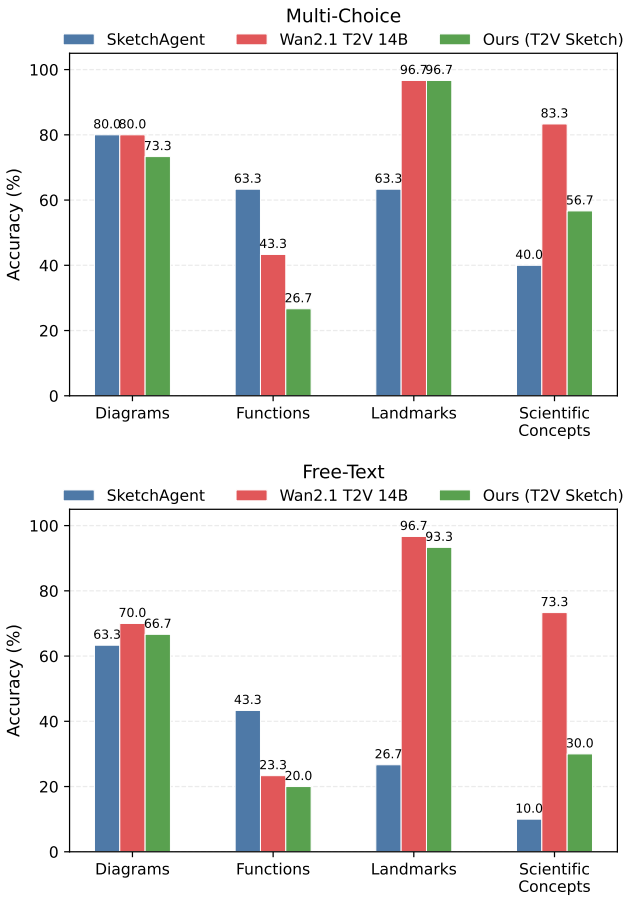

Fig. 21. **Classification accuracy across categories.** Video-based methods excel on Landmarks (97% vs. 63.3% for SketchAgent), while SketchAgent dominates Functions due to its LLM backbone. Our method's accuracy closely tracks Wan 2.1, confirming that both knowledge and limitations transfer from the base model.

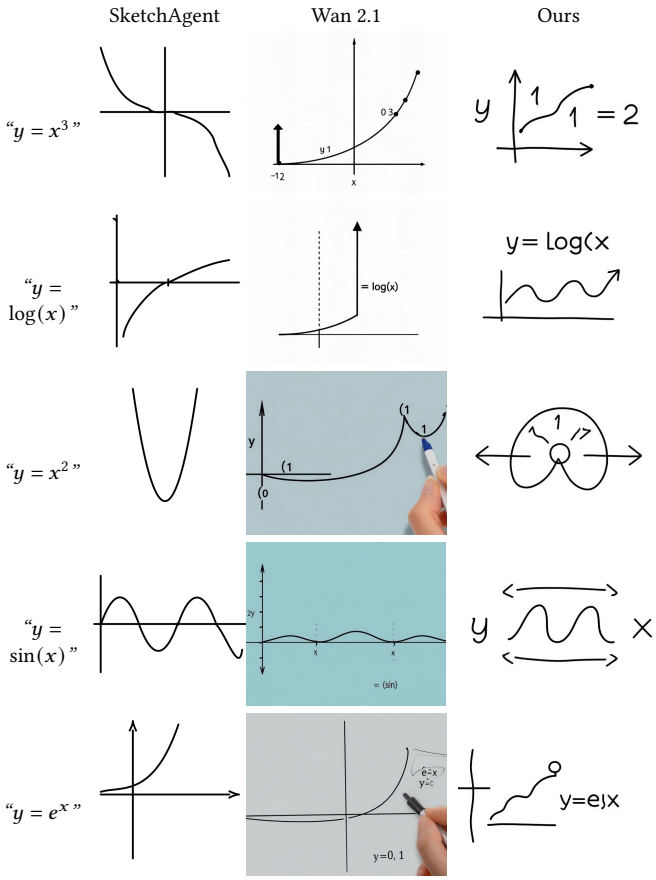

Fig. 22. **Functions.** Representative results for mathematical functions. This category most clearly demonstrates the advantage of LLM-based approaches: SketchAgent produces precise, mathematically correct curves due to its language model backbone. Both Wan 2.1 and our method struggle—curves are often incorrect or unrecognizable, and text/equations are garbled (e.g., $y = log(x)$ in our output). This reveals a fundamental limitation of video models for concepts requiring symbolic or mathematical knowledge.

We evaluate this behavior on two sets: 100 QuickDraw object concepts, which can be considered relatively simple (e.g., cake, ice cream, pants), and 40 scene-level prompts, which are more complex (e.g., a Paris street at dusk). For each concept, we generate two sketch videos with different random seeds, resulting in 200 QuickDraw videos and 80 scene videos in total. As shown in Table 3, multi-stroke behavior occurs less frequently for QuickDraw concepts than for scene-level prompts, suggesting that increased concept complexity — typically requiring a larger number of strokes — makes this phenomenon more pronounced. Exploring models that operate over longer video lengths, as well as mechanisms that more strongly encourage stroke continuity in pixel space, may help distribute stroke generation more evenly over time and mitigate this effect, particularly for complex scenes.

C Interactive Sketching Interface

Figure 24 shows a demo of our interactive sketching interface. We built a prototype collaborative interface for the autoregressive model, where the user and the model co-draw on a shared canvas, given a text prompt. The interaction is turn-based: the user can add (or erase) strokes, then the model continues the sketch by predicting the next sequence of strokes conditioned on the current canvas. This enables real-time, incremental refinement and allows the model to adapt to user edits on-the-fly.

The interface exposes core generation parameters such as resolution, number of frames per run, overlap between consecutive runs, and random seed. At first run, the concept from the user input will be refined to a detailed prompt automatically by an LLM. Each press of the *Run* button generates a short continuation segment; overlap frames refers to context frames obtained from the previous run, which are used to stitch segments smoothly while preserving existing content. Users can switch between brush and eraser tools to modify the canvas between runs, enabling iterative co-creation with the model.

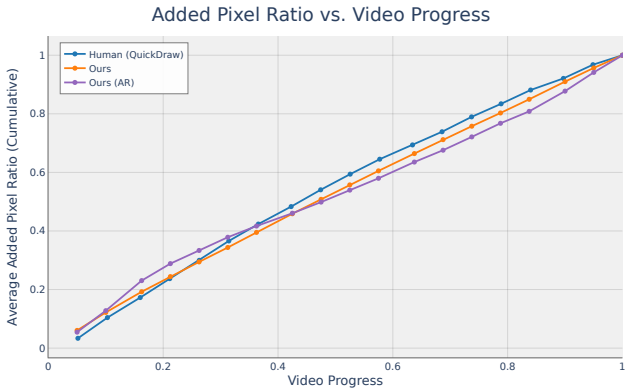

Fig. 23. **Accumulated ratio of newly added pixels as a function of video progress.** Values are normalized such that the final frame equals 1. Our method exhibits a smooth and steady accumulation curve, reflecting incremental stroke additions over time and closely mirroring human drawing behavior.

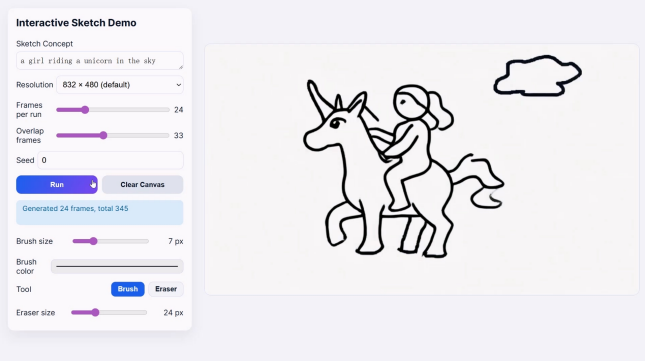

Fig. 24. **Demo of our interactive sketching interface.** Users can co-draw with the model on the shared canvas for a concept.

D Prompt Adherence

We evaluate prompt adherence by progressively enriching an initially simple text instruction with additional details. As shown in Figure 25, the model follows the updated prompt by adding the newly requested elements (e.g., doors, chimney, windows, fence) while maintaining the previously drawn structure, demonstrating compositional control over the sketching process.

E Training Data

We visualize the training data used for training our two-stage model. In Figures 26 and 27 we provide exemplars of the simple geometric primitives used in the first stage. We also provide the seven real human sketches used in our second stage in Figure 28.

F Additional Qualitative Results

Text-to-Video Results. We provide additional sketch results of our text-to-video model in Figures 29 and 30, covering diverse prompts and instructions. These examples highlight clean, high-quality, and semantically meaningful sketches with incremental drawing progression over time.

Image-to-Video Results. We show image-to-video model results in Figures 31 to 34. Conditioning on the first-frame brush exemplar enables faithful transfer of color and texture across the full sequence, including brush styles not seen during training.

Autoregressive Results. We include additional sketches of our autoregressive model in Figures 35 and 36. These sequences preserve coherent, stroke-by-stroke progression, although with slightly reduced visual fidelity compared to our diffusion-based models.

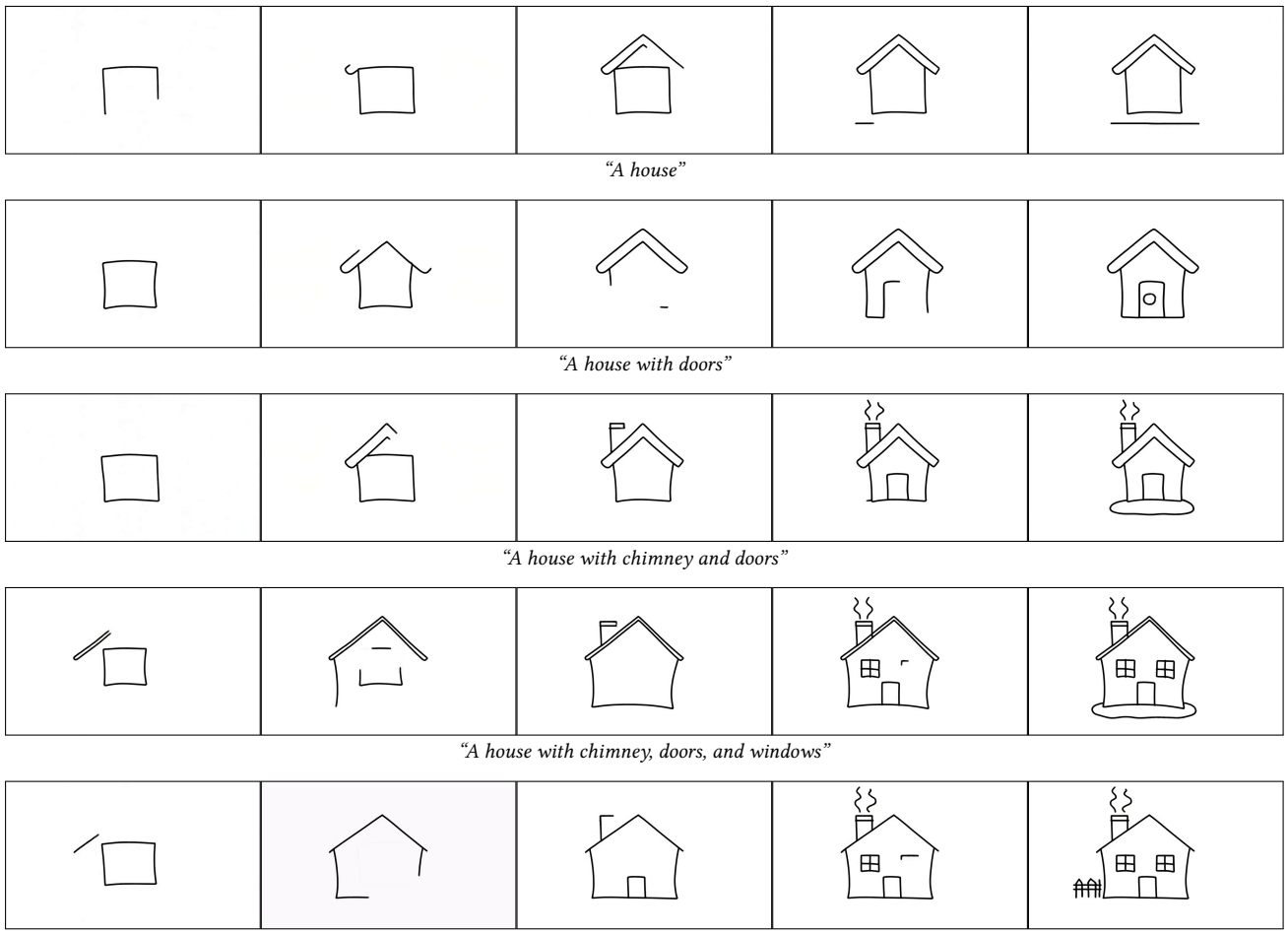

Fig. 25. **Prompt adherence with incremental details.** Given a base concept (*a house*), we progressively add details in a prompt (e.g., *with doors*, *with chimney and doors*, *with chimney, doors, and windows*, and *with chimney, doors, windows, and fence*). The generated sketch sequences consistently follow the prompt and incorporate the newly requested elements.

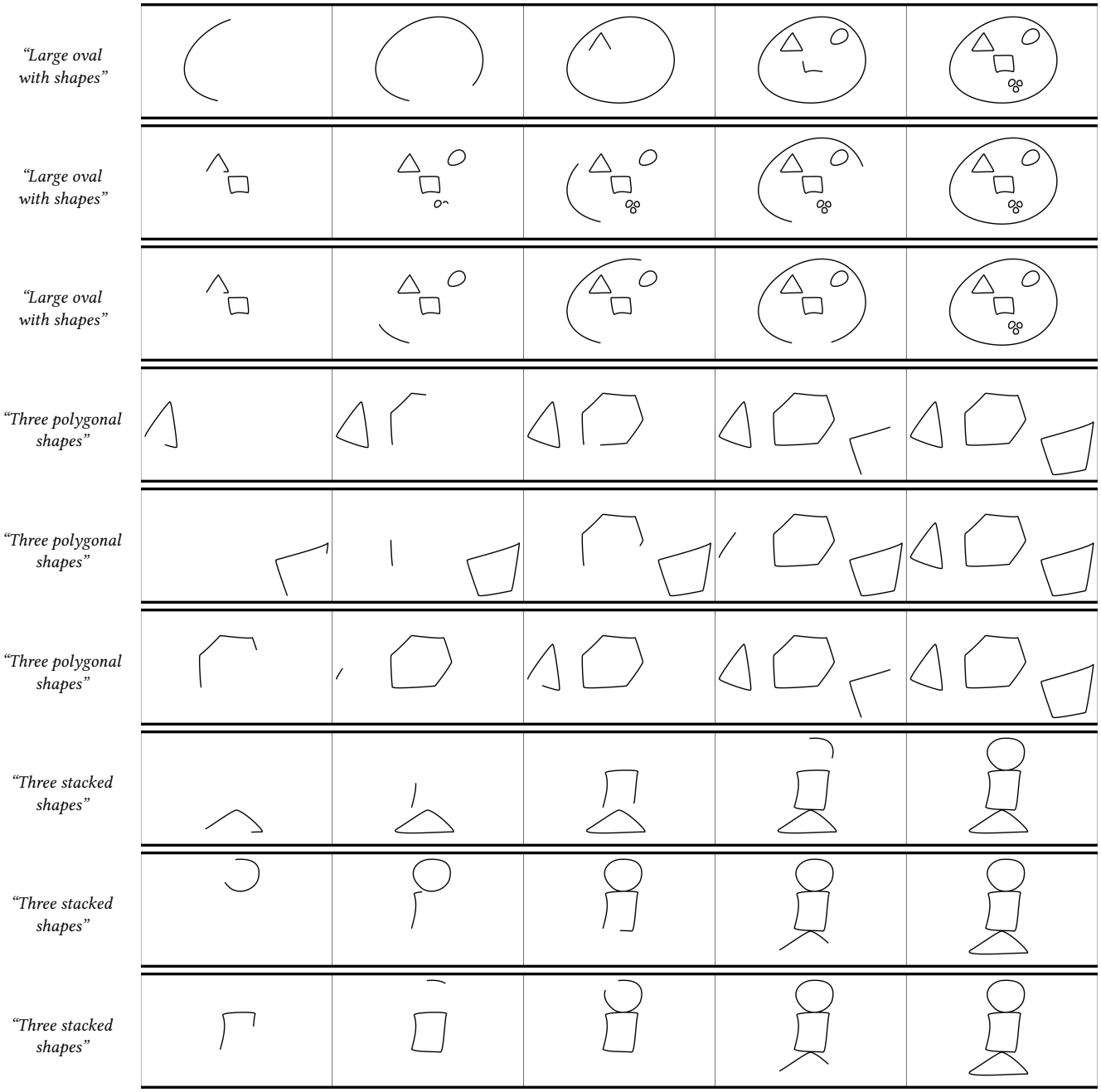

Fig. 26. **Training data exemplars of our simple geometric primitives.** Examples of the simple geometric primitives used in the first stage of our fine-tuning pipeline, focused on teaching the model basic drawing "grammar".

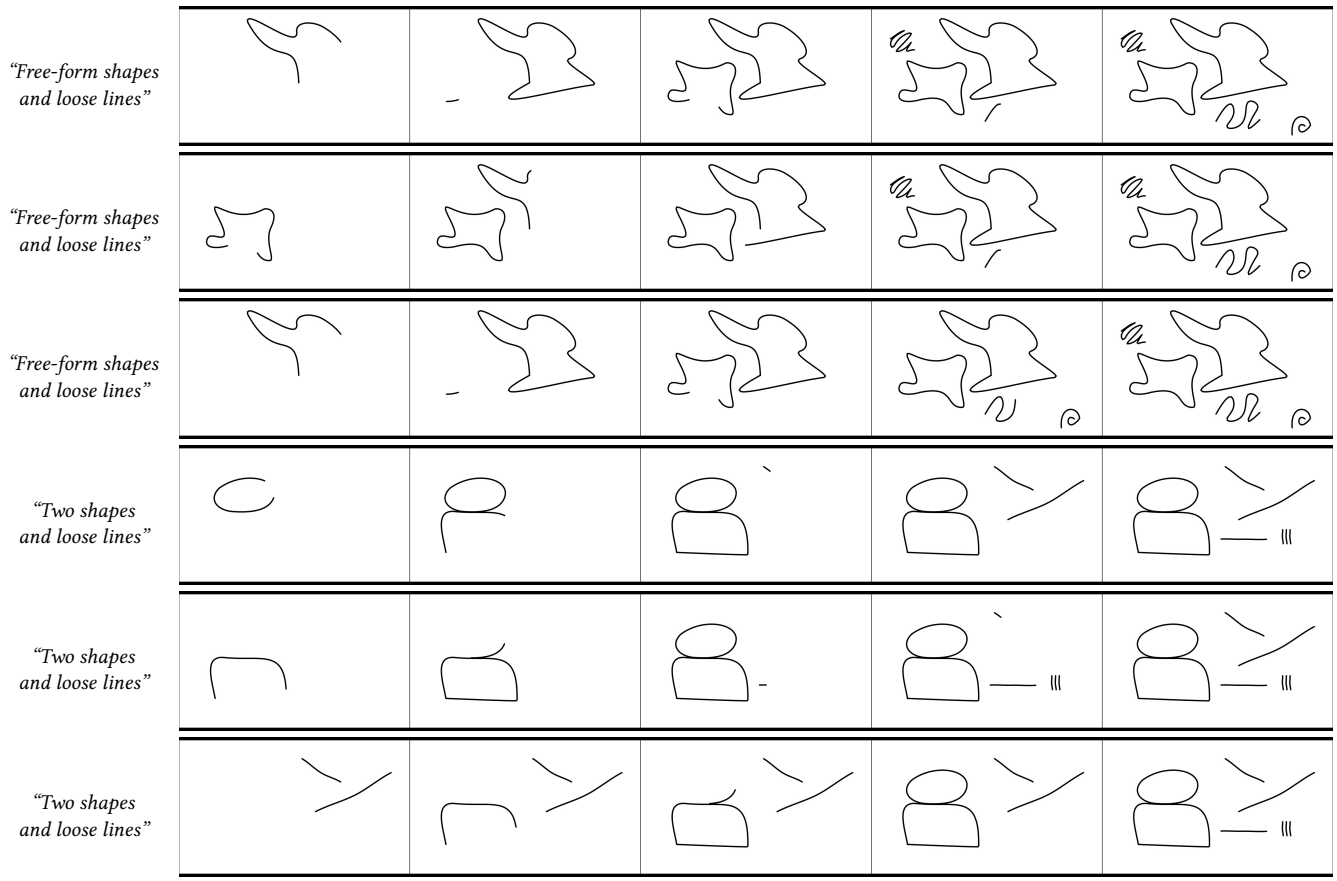

Fig. 27. **Training data exemplars of our simple geometric primitives.** Additional examples of the simple geometric primitives used in the first stage of our fine-tuning pipeline, focused on teaching the model basic drawing "grammar".

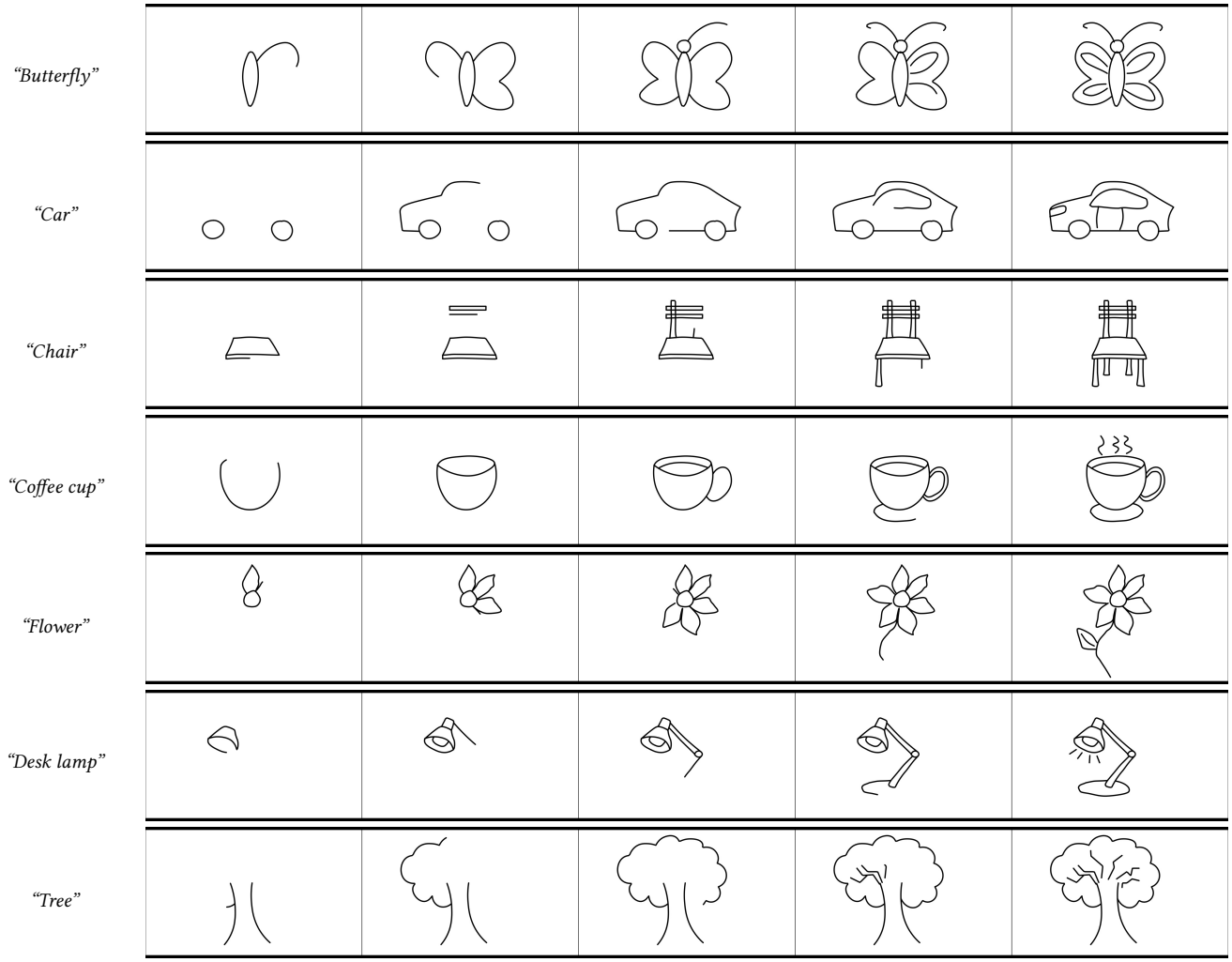

Fig. 28. **Our seven real human sketches used for training.** We provide the seven human-drawn sketches used in the second stage of our fine-tuning pipeline.

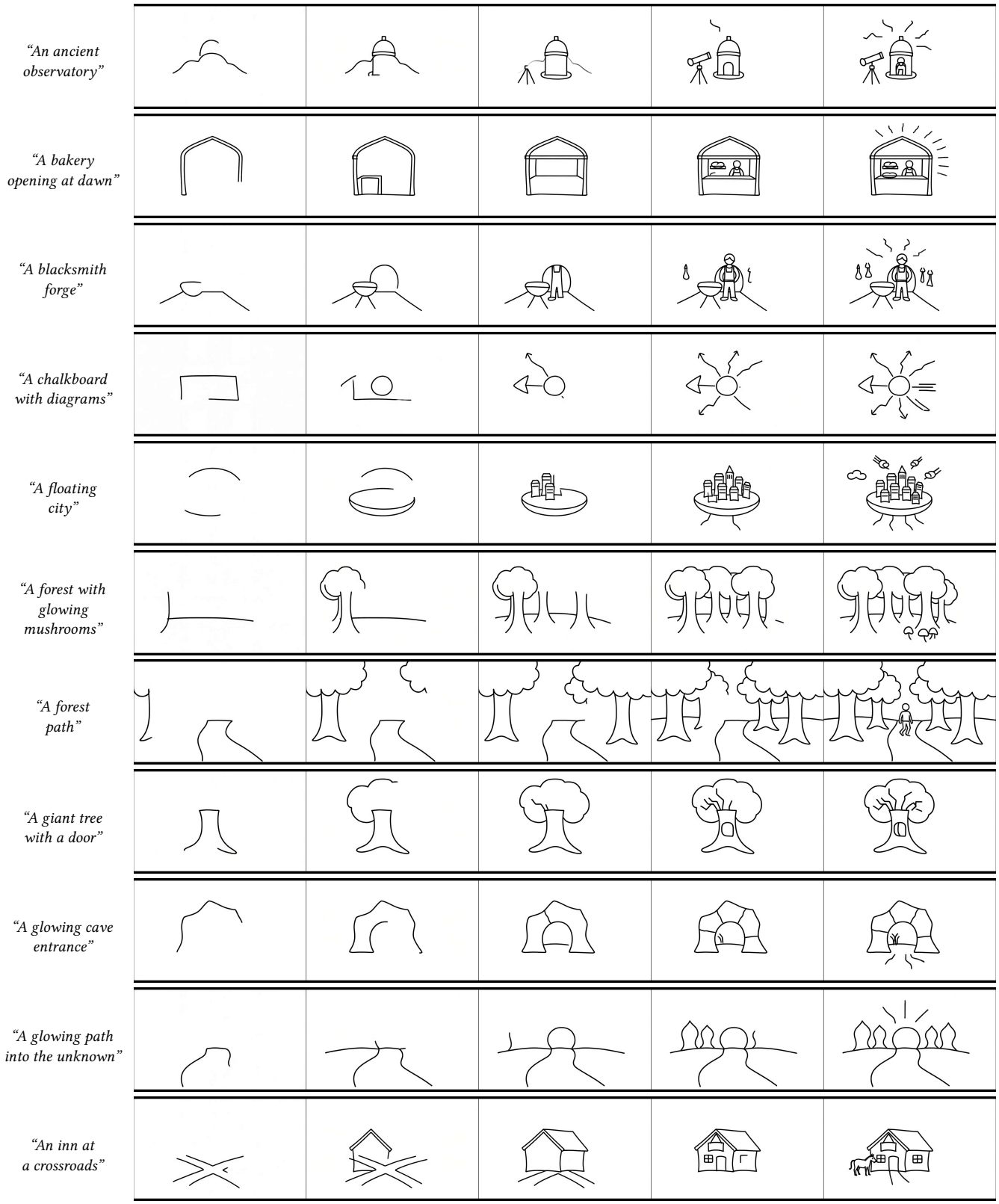

Fig. 29. **Additional text-to-video results (1/2).** Additional qualitative results generated with our text-to-video model.

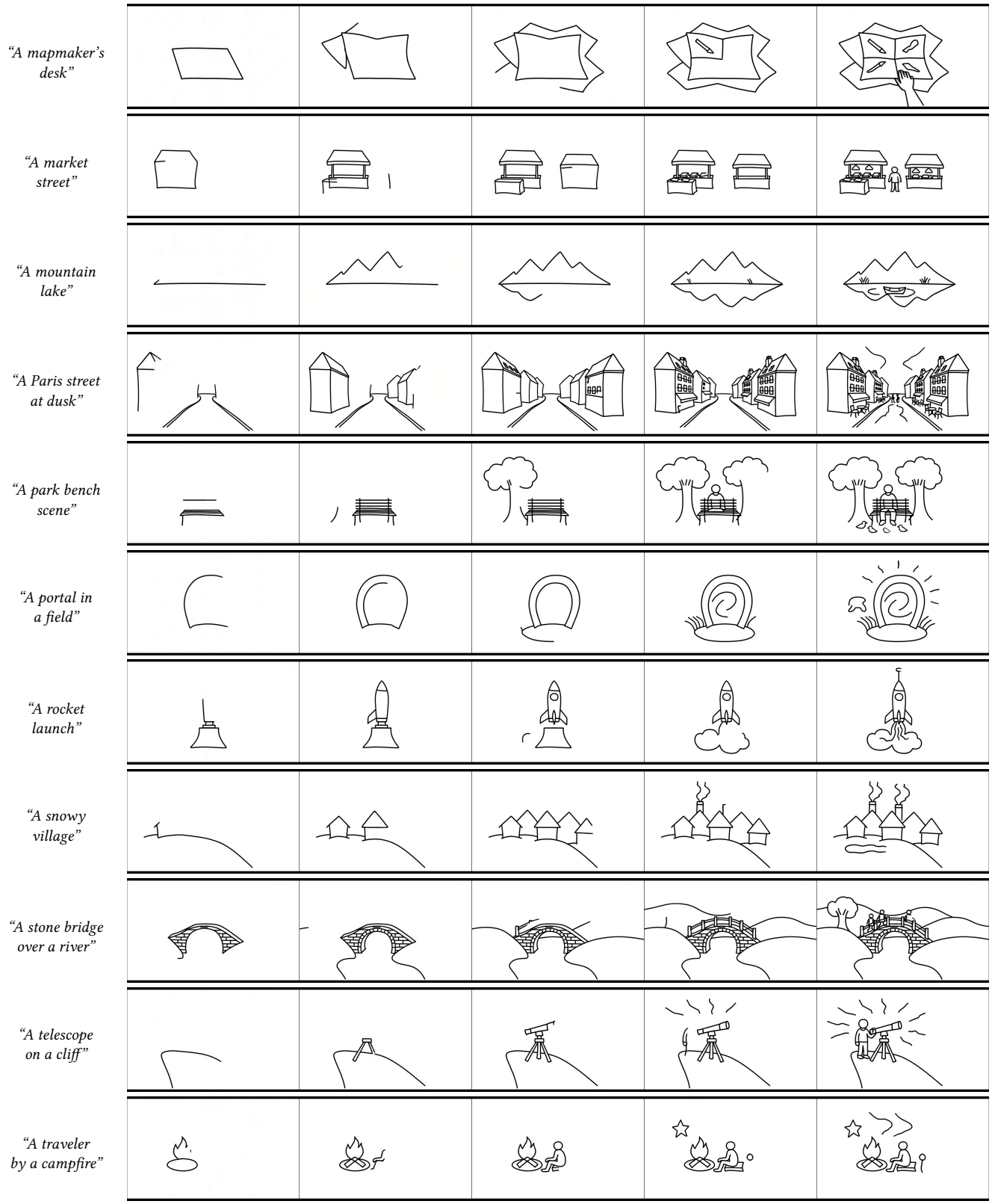

Fig. 30. **Additional text-to-video results (2/2).** Additional qualitative results generated with our text-to-video model.

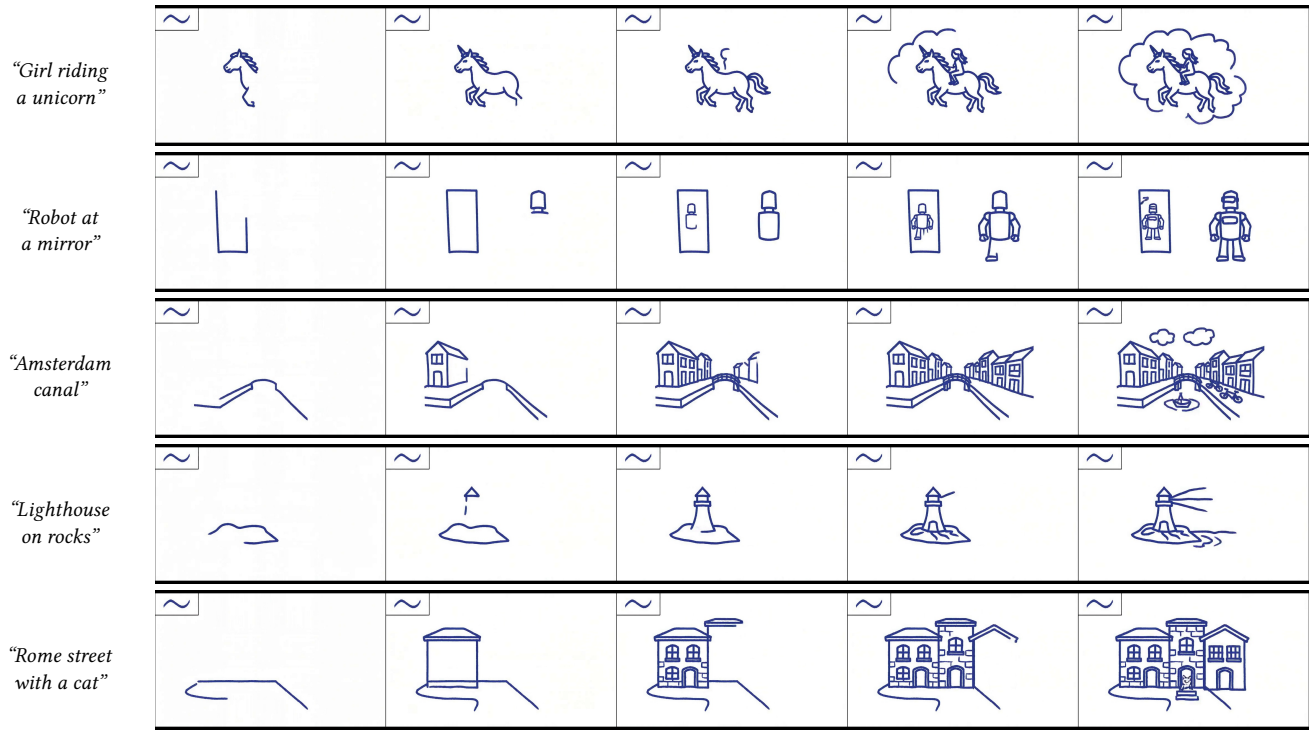

Fig. 31. **Additional brush control I2V results.** We show results using an unseen brush style (*caligraphy-vertical*) and color (*indigo-blue*).

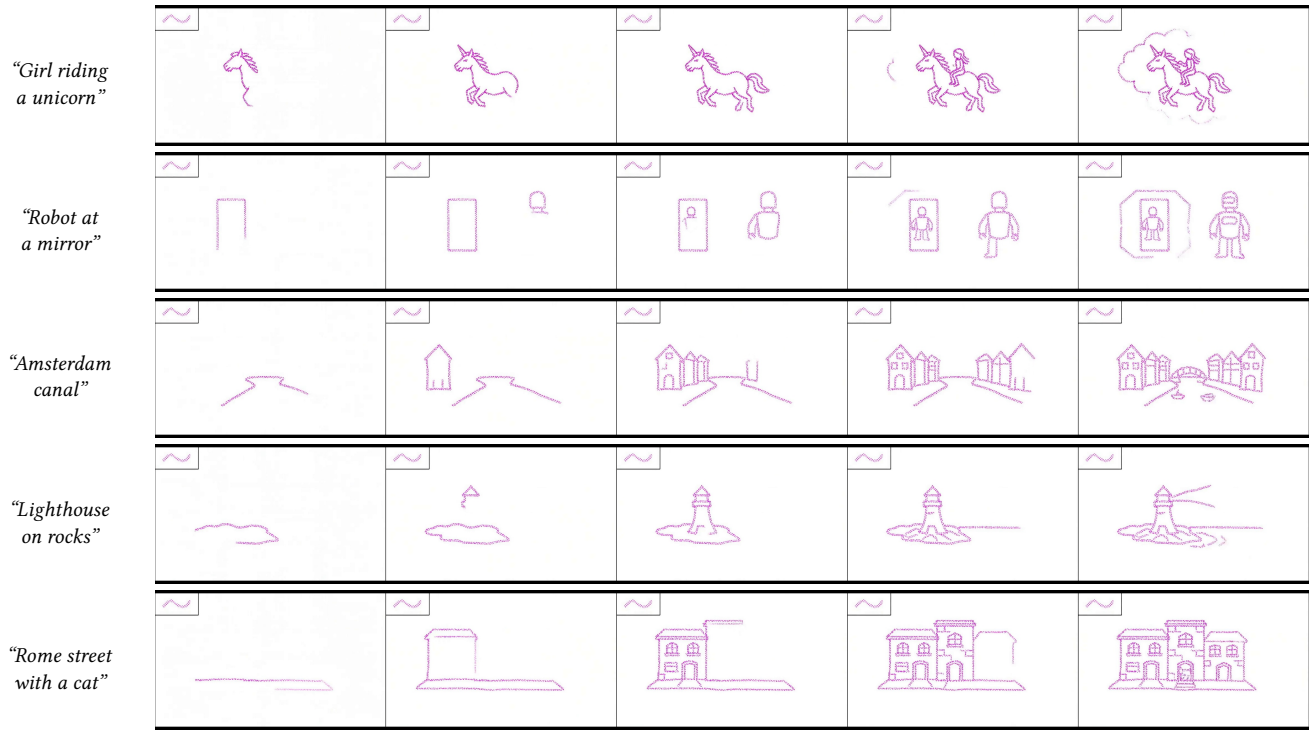

Fig. 32. **Additional brush control I2V results.** We show results using an unseen brush style (*hard-large-dots*) and color (*pink*).

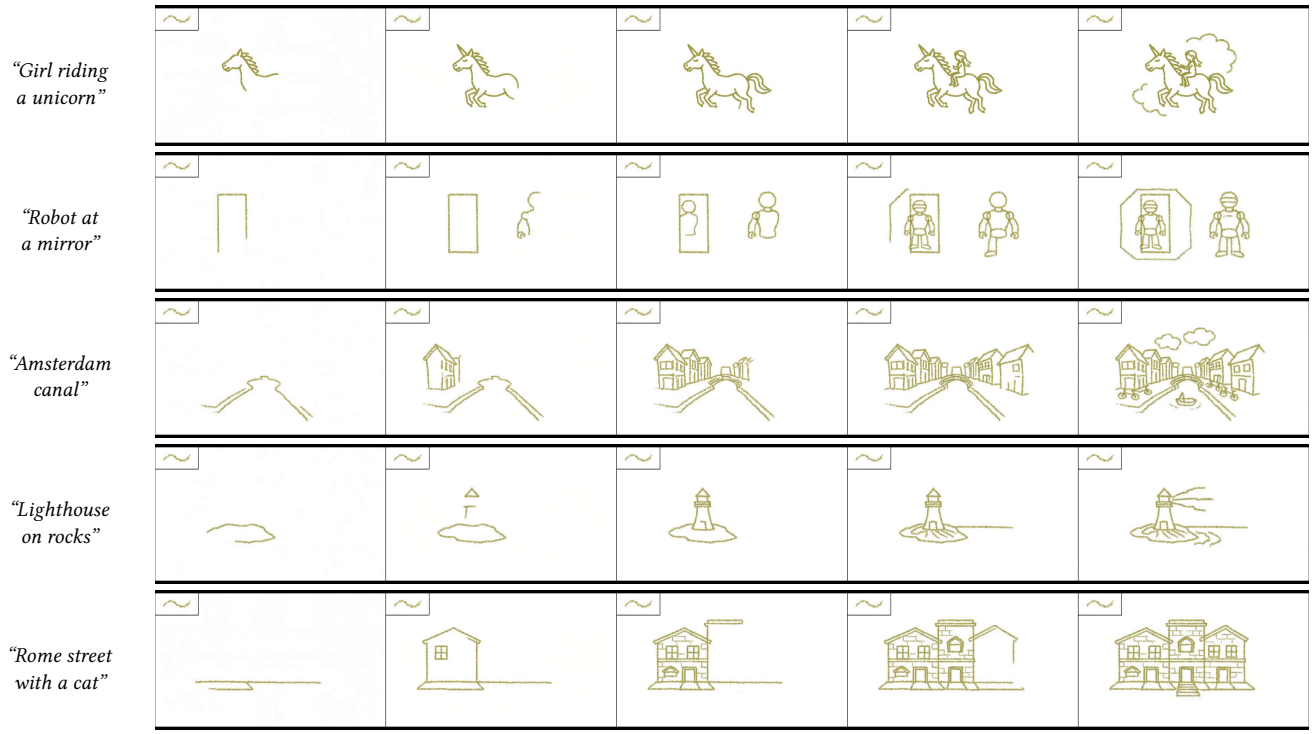

Fig. 33. **Additional brush control I2V results.** We show results using an unseen brush style (*bubbles*) and color (*mustard-olive*).

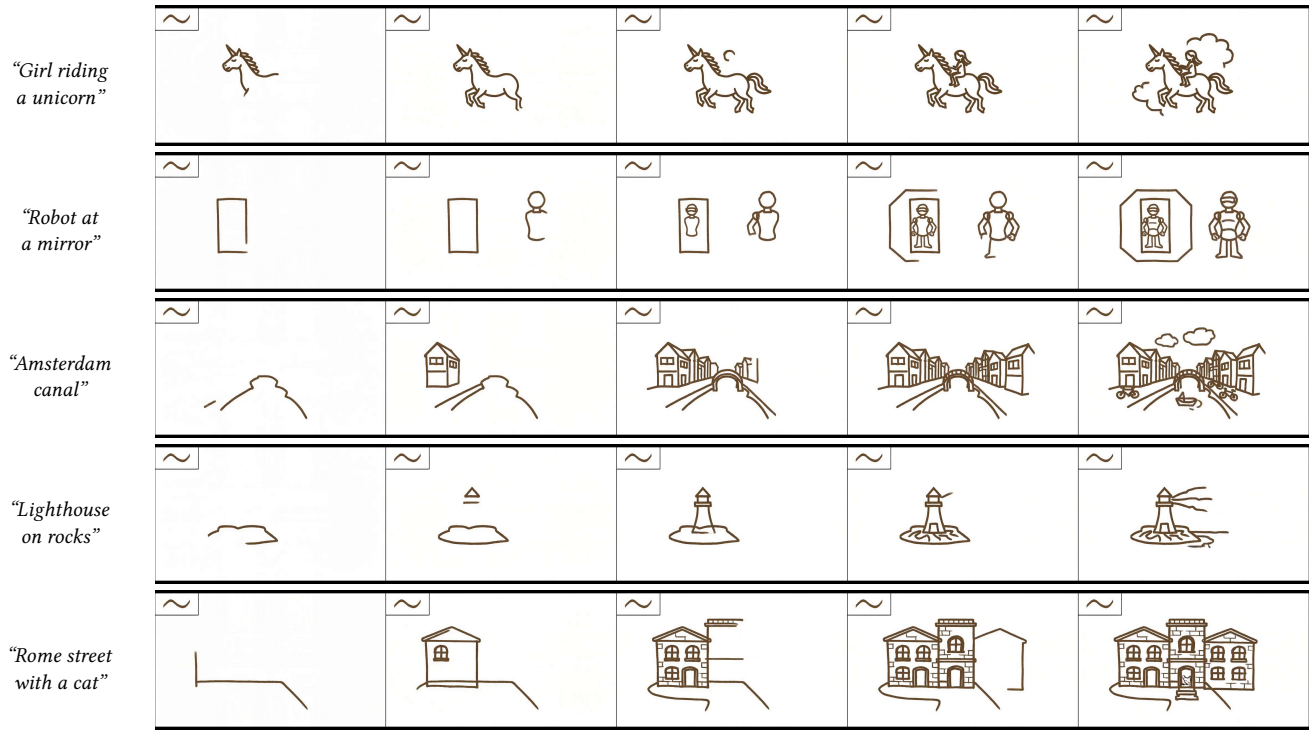

Fig. 34. **Additional brush control I2V results.** We show results using an unseen brush style (*caligraphy*) and color (*mocha-brown*).

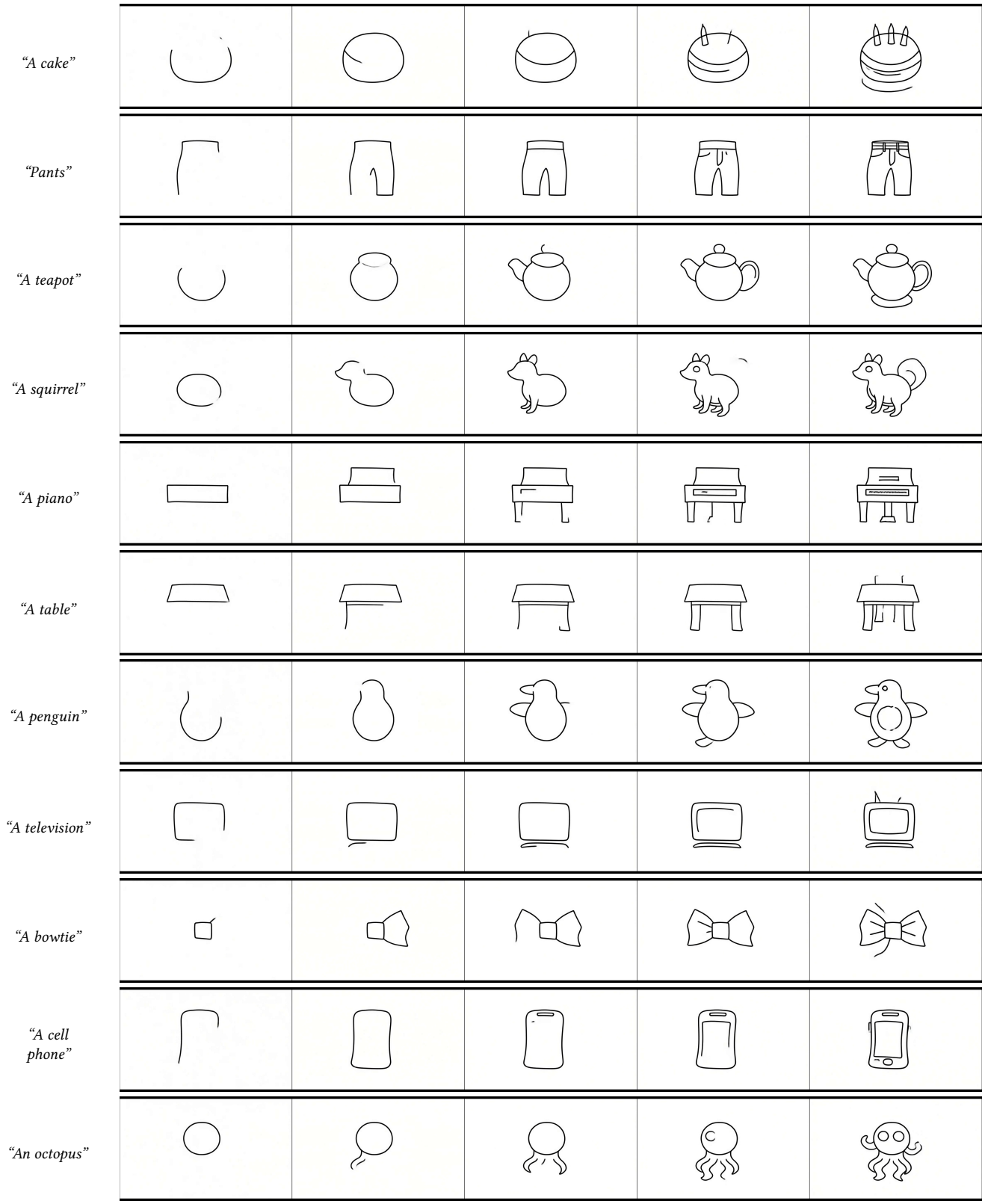

Fig. 35. **Autoregressive results.** Additional autoregressive model results on QuickDraw prompts.

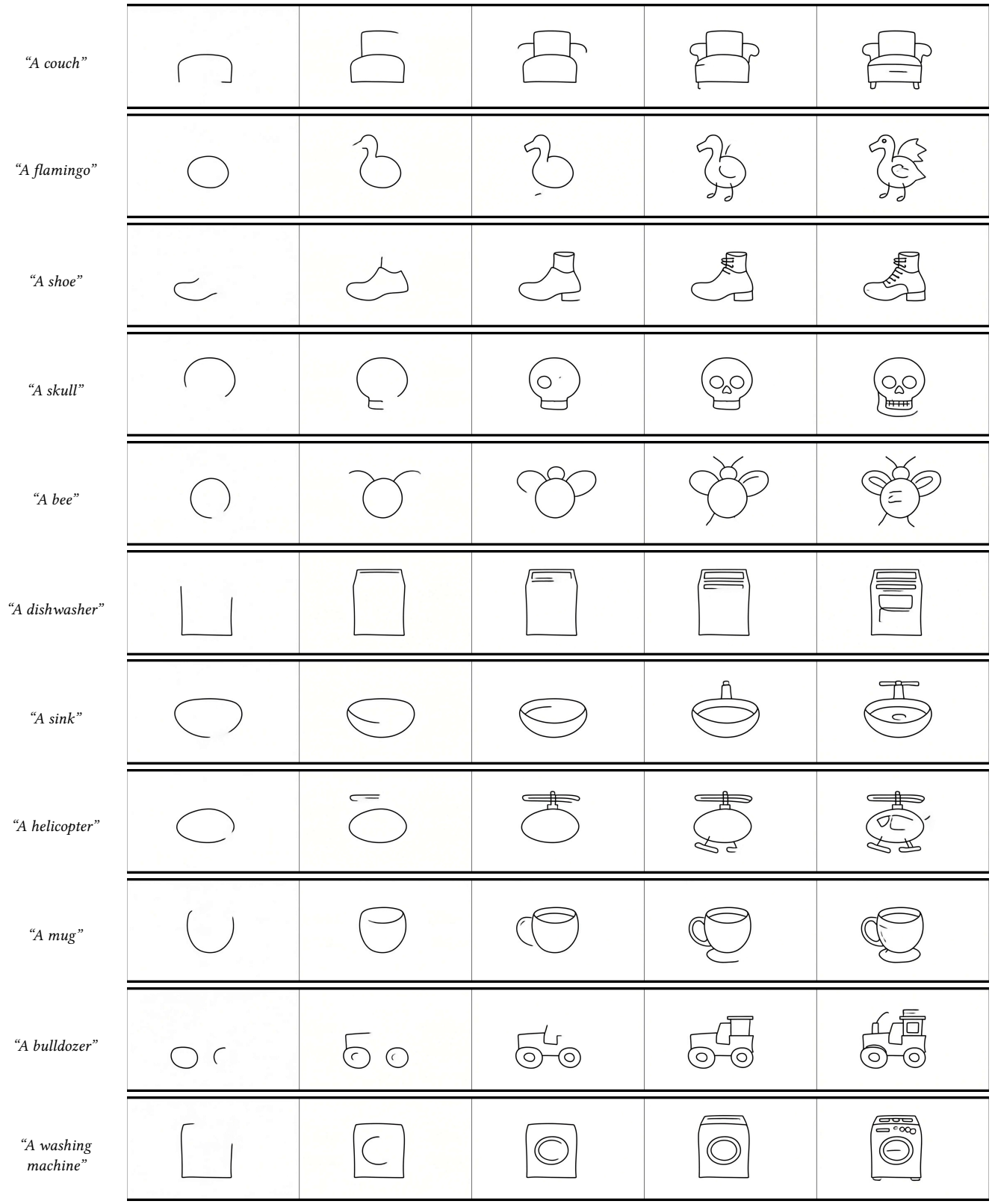

Fig. 36. **Autoregressive results.** Additional autoregressive model results on QuickDraw prompts.