SketchFlow: Zero-Shot Vector Sketch Generation via GMM Prior Flow in CLIP Latent Space

(Supplementary Material)

JIN ZHOU*, HONGLIANG YANG*, PENGFEI XU†, and HUI HUANG, Shenzhen University, China

1 Overview

This supplementary material complements the main paper with the following content: (i) **preprocessing details**, including length-proportional point allocation, short-stroke handling, and final re-sampling; (ii) **implementation details** of **SketchFlow**, covering network architectures, training setup, and inference configurations with prompt examples and hyperparameter guidance; (iii) a **cross-domain transfer evaluation** on TU-Berlin and Creative-Mix with both quantitative and qualitative comparisons; (iv) an **extended zero-shot generation** gallery together with a broader analysis of NeuralSVG under varying stroke counts; and (v) **user study details**, including the questionnaire design, drawing-order visualization, and extended qualitative analysis.

2 Preprocessing Details

We convert each vector sketch to a fixed sequence of $n = 256$ points, yielding a uniform drawing-time parameterization for the decoder. We next describe proportional point allocation and short-stroke handling.

2.1 Length-Proportional Point Allocation

Given a vector sketch containing M strokes $(\mathbf{s}_1, \ldots, \mathbf{s}_M)$, each with an arc length L_k, we employ a length-proportional strategy to distribute the total $n = 256$ points:

(1) **Initial Assignment:** Every stroke $\mathbf{s}_k$ is initially allocated one point.
(2) **Proportional Distribution:** The remaining $N' = n - M$ points are distributed among the strokes proportionally to their arc lengths L_k. The initial allocated point count N_k^{init} for stroke $\mathbf{s}_k$ is:

$$N_k^{\text{init}} = 1 + \text{round}\left(N' \cdot \frac{L_k}{\sum_{j=1}^{M} L_j}\right)$$

where the total points $\sum N_k^{\text{init}}$ is corrected to equal n if rounding leads to a slight discrepancy.

2.2 Handling of Short Strokes

In rare cases, especially for extremely short strokes, the initial allocation may result in a stroke being assigned $N_k^{\text{init}} = 1$ point. To facilitate accurate, uniform resampling (which requires at least two points to define a segment) and better capture the trajectory, we perform a point re-allocation:

(1) We identify any stroke $\mathbf{s}_k$ for which the allocated point count N_k^{init} is exactly 1.
(2) For each such stroke, we borrow one point from the stroke $\mathbf{s}_{\max}$ that currently holds the largest arc length $L_{\max}$ (and $N_{\max} \geq 2$ points). This re-allocation is performed until all strokes have $N_k \geq 2$ or the longest stroke can no longer afford the transfer, ensuring the minimal representation for a trajectory stroke is two points whenever possible.

2.3 Final Resampling and Encoding

After the final point counts N_k are determined, each stroke $\mathbf{s}_k$ is uniformly resampled along its arc length. The segment length for resampling is calculated as $l_k = L_k/(N_k - 1)$. The resulting points from all M strokes are concatenated in their original drawing order to form the sequence $\mathbf{s} \in \mathbb{R}^{n\times 2}$. The final input $\tilde{\mathbf{s}} \in \mathbb{R}^{n\times 3}$, incorporating the pen-state flag p_i, is then constructed as detailed in the main paper. Regarding stroke preprocessing, we scale the binary pen-state indicators to $\{-0.1, 0.1\}$ and treat them as continuous variables. Empirically, we observe that this small variance scaling prevents the diffusion model from prematurely committing to discrete state transitions at early, high-noise generation stages. It effectively encourages the model to prioritize synthesizing the global geometric trajectory before resolving local stroke breaks.

3 Implementation Details

3.1 Network Architectures

For the OT-Flow Matching Model, the velocity field network v_θ is parameterized by a residual MLP equipped with Feature-wise Linear Modulation (FiLM) layers, featuring a hidden width of 1024 and a depth of 10 layers. During training, we apply an L2 regularization penalty on the predicted velocity vectors to minimize the kinetic energy of the flow. For the Hybrid Diffusion Decoder, the 1D U-Net backbone utilizes a base dimension of 128, with channel multipliers set to (1, 1.5, 2, 4). At each resolution scale, we interleave 2 Transformer blocks, where the number of attention heads increases progressively with the network depth (4, 4, 8, 8).

3.2 Training Setup

The OT-Flow Matching model and Hybrid Diffusion Decoder are jointly trained for 1,221,444 optimizer steps (518 epochs, using 20% of the shuffled training loader per epoch). Training takes approximately three days on 8 × NVIDIA RTX 3090 GPUs. We use AdamW with a learning rate of 1e-4, a weight decay of 1e-3, and β values of (0.9, 0.999). The total batch size is 256. For the adaptive prior variance injection, we set the mean standard deviation to 0.025 and introduce

*Jin Zhou and Hongliang Yang contributed equally to this work.
†Corresponding author: Pengfei Xu

Authors' Contact Information: Jin Zhou, doudin2618@gmail.com; Hongliang Yang, hongliang.cscg@gmail.com; Pengfei Xu, xupengfei.cg@gmail.com; Hui Huang, hhzhiyan@gmail.com, Guangdong Provincial Key Laboratory of Visual Media and Multidimensional Intelligence, CSSE, Shenzhen University, Shenzhen, China.

a log-normal perturbation by randomly scaling it with a logarithmic coefficient sampled with a standard deviation of 0.25.

3.3 Inference Configuration

For inference, Heun's method integrates the transport ODE for 60 steps, and the Hybrid Diffusion Decoder then uses 60 DDPM denoising steps. In this section, we provide detailed inference settings and the specific text prompts used to generate the results showcased in the paper, particularly in the teaser figure.

Prompts for Typographic and Symbolic Sketches. For the stylized letters and numbers demonstrated in the teaser, we utilized specific descriptive prompts to guide the structure and style. Examples include:

- "A minimalist logo of the letter G with double-stroke effect"
- "A simple letter s"
- "A symbol of the number 2 with double-stroke effect"

Similar prompts were employed to synthesize the remaining typographic results.

Prompts for Architectural Sketches. For the complex building structures shown in the teaser, we applied the following detailed prompts to ensure structural coherence and specific architectural features:

- "Twin towers."
- "A high tower with a spherical top and a sharp needle spire."
- "A building contains the copper onion-domed clock tower, and the flanking copper-domed minarets."
- "A church building with a central bell turret with a cross, arched three entrance portals."
- "A mosque featuring a large central dome and multiple smaller flanking domes with a prominent central minaret spire."

All other explicitly annotated results in the manuscript were generated using their corresponding provided text prompts.

Hyperparameter Settings. The generation process is highly dependent on two key hyperparameters: the base prior variance σ and the sampling noise scaling factor γ. We empirically established the following default configurations for different generation scenarios:

- **Known categories:** $\sigma = 0.025$, $\gamma = 1.0$
- **Prompts beyond the training vocabulary:** $\sigma = 0.025$, $\gamma = 0.4$
- **Specific action prompts:** $\sigma = 0.04$, $\gamma = 0.4$
- **Image-conditioned generation:** $\sigma = 0.04$, $\gamma = 0.0$

Guidance on Parameter Tuning. We use simple heuristics to adjust σ and γ. The base prior variance σ controls the semantic search space. If outputs are low quality and overly similar, we increase σ; if they vary excessively or lack structural consistency, we decrease it. The scaling factor γ controls the trade-off between stability and diversity. We decrease γ when generation requires high fidelity and strict adherence to the input condition.

4 Cross-Domain Transfer Evaluation

To comprehensively assess the model's cross-domain transfer ability beyond the primary QuickDraw training corpus, we further evaluate **SketchFlow** on three supplementary datasets:

Table 1. **Quantitative comparison of FID on Creative datasets.** Lower scores indicate better visual fidelity and structural similarity to the target distribution.

Method	Creative Birds	Creative Creatures	Creative Mix
Doodleformer	58.118	63.212	58.219
Ours	**34.444**	**32.938**	**32.181**

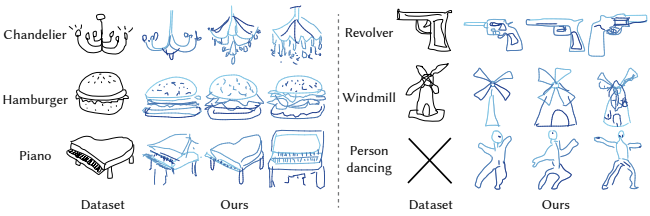

Fig. 1. **Qualitative results on the TU-Berlin dataset.** Our model effectively learns the dataset's unique freehand style and produces coherent sketches, even for categories absent from the original dataset.

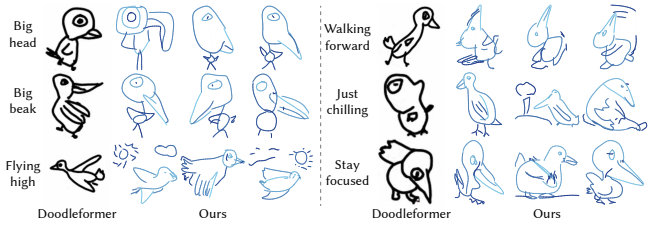

Fig. 2. **Qualitative results on the Creative-Mix dataset.** Our method successfully adapts to the highly creative style while maintaining strict semantic control, visibly outperforming Doodleformer in overall structure and clarity.

- **TU-Berlin Sketch Dataset**: This benchmark contains 250 object categories and roughly 20,000 freehand black-and-white sketches, with approximately 80 sketches per class.
- **Creative Birds** (8,067 sketches) and **Creative Creatures** (9,097 sketches): These subsets of the "Creative Sketch Generation" dataset [Ge et al. 2021] contain imaginative sketches with part-level annotations. We merge them into a single evaluation dataset, denoted as **Creative-Mix**.

Quantitative Results. We conduct quantitative comparisons on the creative datasets, evaluating visual fidelity using the Fréchet Inception Distance (FID) metric. As reported in Table 1, our method substantially outperforms the baseline Doodleformer across all data subsets. This remarkable reduction in FID indicates that our approach is highly capable of synthesizing sketches with superior visual quality and closer distributional alignment to the target domain, even in highly imaginative and abstract contexts.

Qualitative Results. Qualitative visualizations further validate our model's strong transferability and stylistic adaptability. Figure 1 illustrates the generation results on the TU-Berlin dataset. **SketchFlow** successfully captures the dataset's distinctive drawing style and generates structurally coherent sketches, including zero-shot

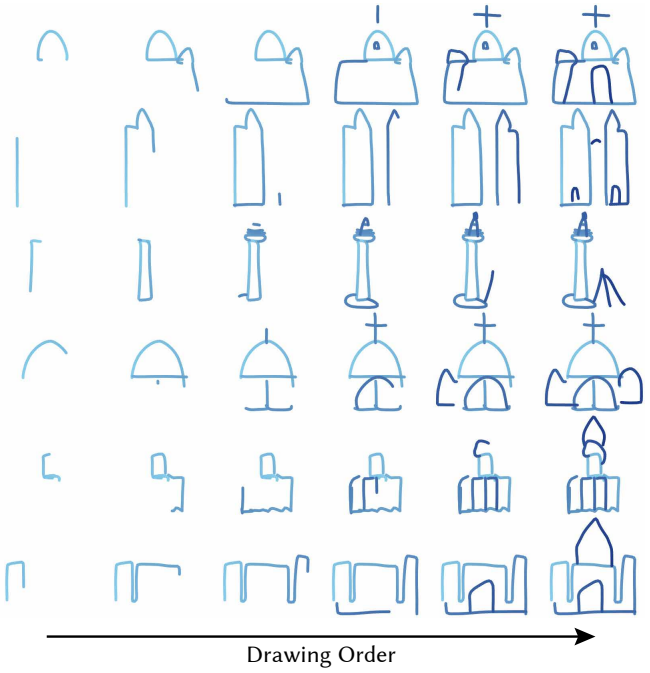

Fig. 3. **Drawing Order Visualization.** To clearly illustrate the sequential nature of our generated vector sketches, stroke trajectories are color-coded according to their generation sequence, progressing from light blue (initial strokes) to dark blue (final strokes). Note that all generated sketches presented in this paper adopt this color-coding scheme to indicate the underlying drawing progression.

examples for categories absent from the primary training corpus. Figure 2 compares results on Creative-Mix. Our method adapts to this highly stylized, imaginative dataset while maintaining semantic control over the target concepts, surpassing Doodleformer in both structural clarity and logical stroke composition.

5 Extended Zero-Shot Generation Beyond the Training Vocabulary

In the main paper, we showed a small number of examples generated from prompts outside the QuickDraw training vocabulary. Figure 5 provides additional results for this setting. The model produces clear sketches for a range of unseen concepts, while multiple samples per prompt demonstrate generation diversity and a recurring hand-drawn style across random seeds. These results demonstrate CLIP-mediated zero-shot generation on the evaluated prompts.

5.1 Unseen-Label Quantitative Evaluation

Table 3 reports the complete per-prompt results for the 44-label evaluation summarized in the main paper. For each prompt, we report the nearest QuickDraw text anchor and its cosine similarity, together with the mean CLIP Score over 32 samples generated without sample selection.

5.2 Comparison with T2I+CLIPasso

We compare **SketchFlow** with a two-stage T2I+CLIPasso pipeline on six single-concept prompts from our unseen-label evaluation: *Pikachu*, *Tetromino*, *Goblin*, *Rocket*, *Potion*, and *Ghost*. For each prompt, SDXL Base 1.0 [Podell et al. 2024] generates one 1024×1024 target using 50 denoising steps, classifier-free guidance 5.0, and fixed seed 2026, without a refiner or reranking. We then run the official CLIPasso implementation [Vinker et al. 2022] with 16 strokes and 2,001 optimization iterations for seeds 0, 1000, and 2000, retaining the result with the lowest CLIPasso evaluation loss.

5.3 Additional Analysis on NeuralSVG

NeuralSVG [Polaczek et al. 2025] supports controlling its stroke count during optimization. The main paper reports the 32-stroke setting; here we evaluate 4, 8, 16, and 32 strokes. Table 2 shows the resulting trade-off: fewer strokes reduce RDP but degrade CLIP alignment, while additional strokes improve CLIP gradually and sharply increase RDP complexity. At 32 strokes, NeuralSVG approaches our CLIP score but retains a much higher RDP value. Our method therefore achieves a more favorable balance between semantic fidelity and geometric efficiency in these comparisons.

6 User Study Details and Extended Analysis

As outlined in the main text, we conducted a user study to evaluate the perceptual quality and human-likeness of **SketchFlow**. In this supplementary section, we detail the specific questionnaire provided to participants and offer an extended qualitative analysis of the comparative results.

6.1 Questionnaire Design

Given the dataset's distinct characteristics and the generation setting beyond a fixed training vocabulary, we refined the evaluation questionnaire to focus heavily on abstraction and structural rationality. Participants were asked to evaluate the generated samples based on the following five specific questions:

(1) **Abstraction Fidelity:** Which sketch better aligns with the characteristics of an abstract sketch (rather than a traced image)?
(2) **Semantic Alignment:** Which sketch best resembles the target text prompt?
(3) **Human-likeness:** Which sketch appears more naturally drawn by a human hand?
(4) **Stroke Rationality:** Which sketch features a more logical and coherent stroke arrangement?
(5) **Aesthetic Quality:** Which sketch is globally more visually appealing?

6.2 Drawing Order Visualization

Our framework intrinsically synthesizes sequential vector trajectories. To effectively illustrate the temporal drawing order of the generated sketches throughout this paper, we adopt a color-coding visualization strategy. As depicted in Figure 3, the temporal progression of individual strokes is represented by varying color intensities, transitioning smoothly from light blue for the initial strokes to dark blue for the final ones. This step-by-step decomposition clearly

Table 2. Additional quantitative comparisons on the QuickDraw dataset (*Bus*, *Cat*, *Chair*, and *Train*), specifically evaluating NeuralSVG with varying stroke counts. Bold and underlined values indicate the best and second-best performance, respectively.

	Bus		*Cat*		*Chair*		*Train*	
Method	CLIP↑	RDP↓	CLIP↑	RDP↓	CLIP↑	RDP↓	CLIP↑	RDP↓
NeuralSVG-4	0.208	**36.891**	0.212	**36.374**	0.247	29.980	0.213	**33.941**
NeuralSVG-8	0.215	<u>72.218</u>	0.228	<u>70.475</u>	0.271	58.554	0.215	<u>66.099</u>
NeuralSVG-16	0.219	142.376	0.248	137.232	<u>0.291</u>	115.030	0.221	129.941
NeuralSVG-32	0.224	278.446	**0.261**	267.091	**0.304**	225.663	0.225	256.168
Ours (γ=0.60)	**0.283**	86.310	**0.261**	83.720	0.290	**21.240**	**0.259**	87.140
Ours (γ=0.80)	<u>0.279</u>	89.080	0.255	87.980	0.282	<u>27.220</u>	<u>0.253</u>	91.700
Ours (γ=1.00)	0.270	94.370	0.248	89.090	0.276	36.360	0.245	97.690

Table 3. **Complete unseen-label quantitative evaluation.** CLIP Score is used as a label-alignment proxy and averaged over 32 samples per prompt without sample selection. The nearest anchor and cosine similarity are computed in CLIP text space over the 345 QuickDraw training labels. Prompts are arranged in two side-by-side blocks for compact presentation. Bold indicates the best result among the three evaluated methods.

Prompt and nearest anchor			Mean CLIP Score			Prompt and nearest anchor			Mean CLIP Score		
Prompt	Anchor	Cos.	**SketchFlow**	G. Prior	Interp.	Prompt	Anchor	Cos.	**SketchFlow**	G. Prior	Interp.
Tetromino	square	0.8021	**0.2740**	0.1997	0.2242	Tie	toe	0.8861	**0.2339**	0.1936	0.1875
Pikachu	banana	0.8369	**0.2354**	0.1800	0.1840	Box	book	0.8943	**0.2770**	0.2176	0.2139
Dancing-human	hand	0.8451	**0.2680**	0.1870	0.1959	a stretching cat	cat	0.8971	**0.2529**	0.2248	0.2274
Slime	brain	0.8534	**0.2310**	0.1993	0.1953	Ghost	saw	0.8993	**0.2449**	0.1909	0.1892
Goblin	brain	0.8537	**0.1961**	0.1770	0.1778	a jumping cat	cat	0.9035	**0.2632**	0.2234	0.2423
Kirby	pig	0.8570	**0.2658**	0.2022	0.1915	Earth	saw	0.9063	**0.2395**	0.2065	0.2057
Mario	angel	0.8620	**0.2001**	0.1672	0.1753	Demon	dragon	0.9072	**0.2107**	0.1839	0.1835
Sonic	saw	0.8688	**0.2420**	0.1892	0.1941	Heart	brain	0.9078	**0.2526**	0.1937	0.2034
Mickey	mouse	0.8842	**0.2738**	0.1954	0.1999	Spiral	circle	0.9116	**0.2764**	0.2020	0.1980
Kinabalu	mountain	0.7458	**0.2498**	0.1888	0.1932	Star of David	star	0.9121	**0.2804**	0.2369	0.2238
Smiley sunflower	smiley face	0.8537	**0.2691**	0.1785	0.1815	a running cat	cat	0.9121	**0.2770**	0.2269	0.2419
Zombie	angel	0.8667	**0.2030**	0.1901	0.1838	Sun & cloud	cloud	0.9133	**0.2344**	0.2038	0.1987
White	star	0.8669	**0.2247**	0.2064	0.2109	Ship	cruise ship	0.9155	**0.2755**	0.2218	0.2278
Pine	tree	0.8690	**0.2584**	0.2192	0.2197	Angry face	face	0.9156	**0.2599**	0.2072	0.2080
Letter A	key	0.8698	**0.2488**	0.2245	0.2222	Fire	saw	0.9175	**0.2617**	0.1945	0.1986
Potion	brain	0.8747	**0.2105**	0.1943	0.1892	Water	river	0.9186	**0.2418**	0.2236	0.2243
Torch	lighter	0.8799	**0.2530**	0.1993	0.2054	Moon & star	moon	0.9343	**0.2394**	0.1950	0.1897
Black	saw	0.8814	**0.2143**	0.1854	0.1900	Fear face	face	0.9352	**0.2230**	0.2095	0.2132
Pizza clock	pizza	0.8818	**0.2991**	0.2313	0.2133	Happy face	smiley face	0.9365	**0.2642**	0.2450	0.2500
Twins	saw	0.8827	**0.2056**	0.1522	0.1422	Sad face	face	0.9366	**0.2580**	0.2094	0.2017
Rocket	star	0.8834	**0.2393**	0.1818	0.1782	Two eyes	eye	0.9372	**0.2395**	0.2087	0.2203
Egg	brain	0.8852	**0.2798**	0.2264	0.2194	Lightning cloud	lightning	0.9487	**0.2647**	0.2117	0.2152
Overall average			**SketchFlow**: **0.2480**			Gaussian Prior: 0.2024			Interp CLIP: 0.2034		

demonstrates that **SketchFlow** captures not only the global geometric structure of the target concepts but also learns a coherent drawing sequence that aligns well with plausible human drawing patterns.

6.3 Extended Qualitative Analysis

While the quantitative results demonstrate a strong overall preference for **SketchFlow**, examining the qualitative traits of the baseline methods provides further context for these scores.

Specifically, successful generation results from NeuralSVG tend to exhibit a dense, image-like quality. They are often characterized by numerous short Bézier curves that frequently overlap at the edges, resulting in a visually dense appearance that deviates from concise human abstraction. Conversely, while SketchAgent produces semantically clear outputs, its generated trajectories often appear highly rigid, symmetric, or repetitive, which noticeably diminishes their perceptual human-likeness.

The strong user preference for **SketchFlow** across all measured metrics is consistent with two traits of our trajectory-aware diffusion approach. First, the generated sketches naturally incorporate authentic drawing nuances (*e.g.*, slight stroke variations), leading to superior human-likeness scores. Second, the evaluated results show abstract expressiveness for challenging, amorphous concepts such as "water" and "fire". In these examples, the model conveys an abstract impression without being tied to the contours of a specific reference image.

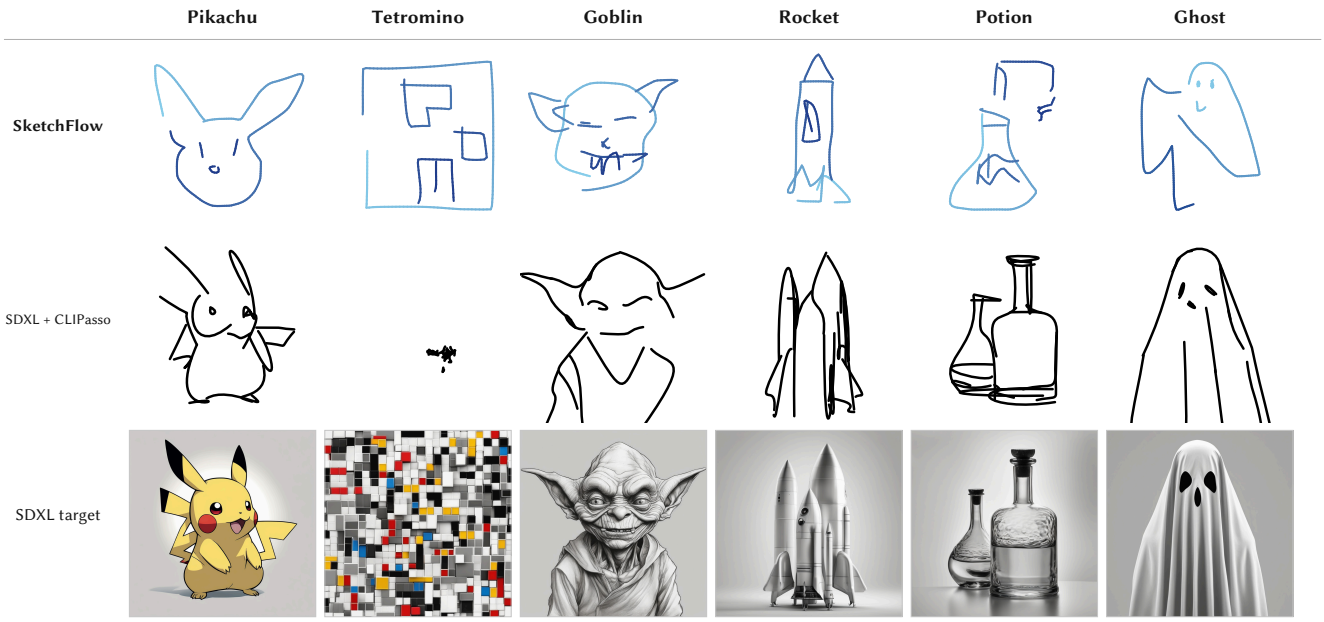

Fig. 4. **Comparison with the two-stage SDXL+CLIPasso baseline.** The **SketchFlow** row reuses samples already presented in the paper; light-to-dark blue indicates drawing order. The bottom row shows the intermediate SDXL targets supplied to CLIPasso. **SketchFlow** prioritizes abstract, concept-defining strokes over perspective and volumetric realism, whereas T2I+CLIPasso often preserves more plausible 3D structure from its raster target but may lose simple visual cues essential to conveying the concept.

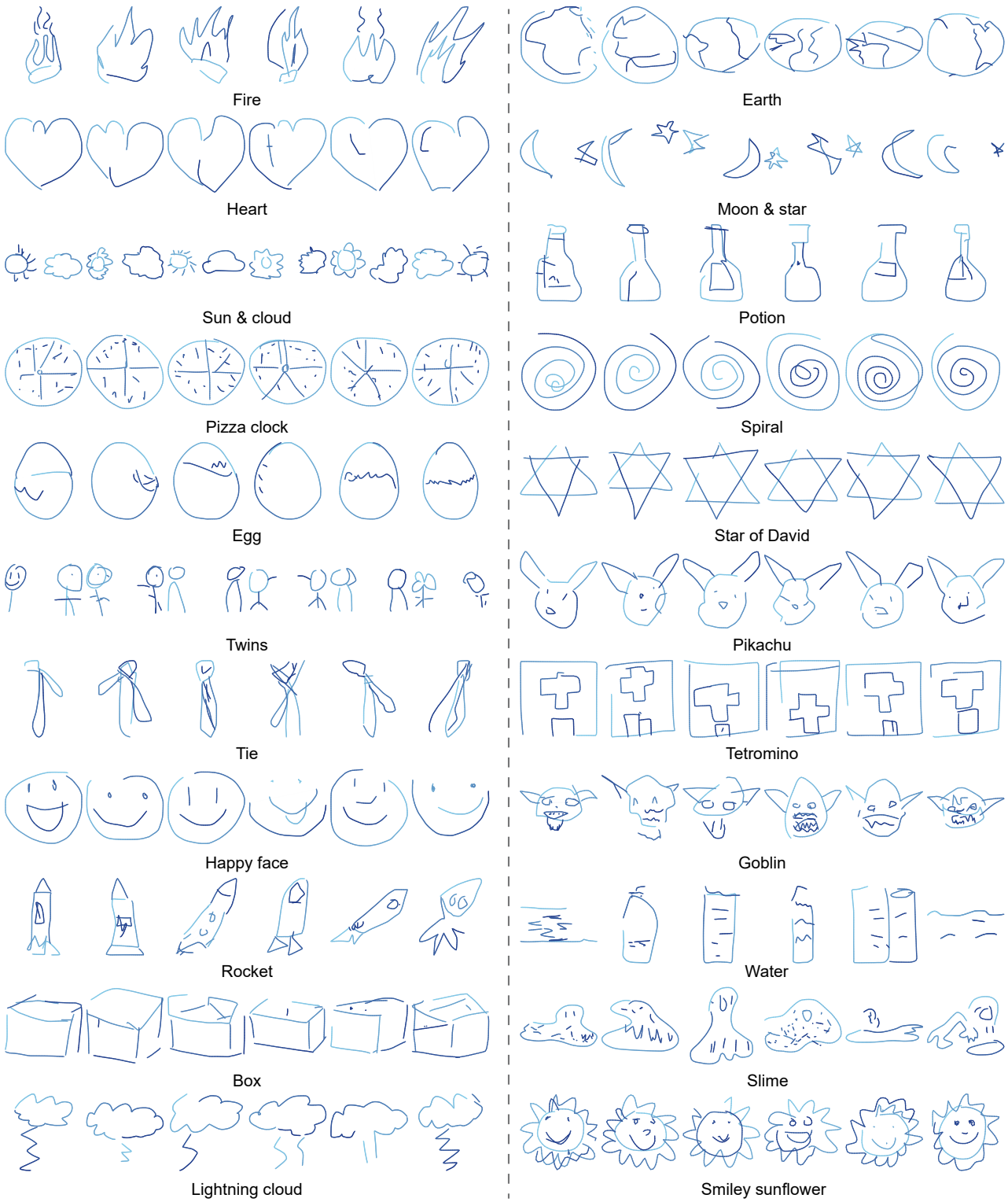

Fig. 5. Extended zero-shot sketch generation beyond the training vocabulary. We show clear, visually appealing sketches for multiple prompts, with samples demonstrating generation diversity and a recurring hand-drawn style across random seeds.

References

Songwei Ge, Vedanuj Goswami, C Lawrence Zitnick, and Devi Parikh. 2021. Creative sketch generation. In *ICLR*.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. 2024. SDXL: Improving Latent Diffusion Models for High-Resolution Image Synthesis. In *ICLR*.

Sagi Polaczek, Yuval Alaluf, Elad Richardson, Yael Vinker, and Daniel Cohen-Or. 2025. Neuralsvg: An implicit representation for text-to-vector generation. In *ICCV*. 15458–15468.

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 TOG* 41, 4 (2022), 1–11.