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

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

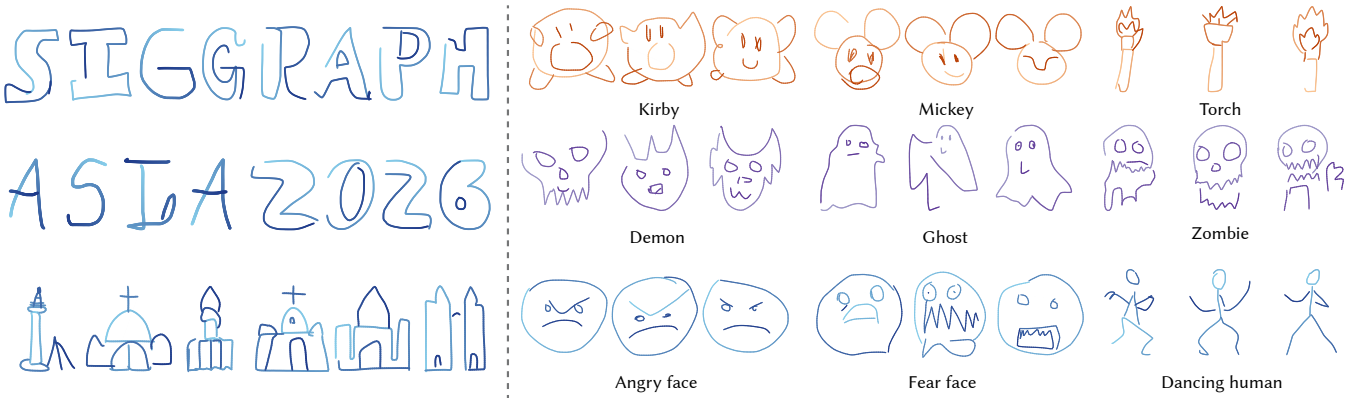

Fig. 1. **Local zero-shot vector sketch generation by SketchFlow. SketchFlow** leverages drawing priors from the CLIP latent space to synthesize prompts beyond its 345-class QuickDraw training vocabulary. Left: Text-prompted generation of letters, numbers, and landmarks in Malaysia. Right: Diverse zero-shot generation including pop-culture characters, creatures, abstract emotions, and dynamic poses. **SketchFlow** produces visually appealing sketches with human-like drawing behaviors without category-specific training examples for these prompts.

Vector sketches remain one of the most concise and immediate mediums for abstract human expression. However, generating high-quality vector strokes that exhibit human-like drawing styles remains an open challenge due to the severe scarcity of fine-grained, high-quality text-to-sketch paired data. Existing text-conditioned generation methods often rely on unstable, time-consuming optimization or struggle to generalize to unseen categories in a zero-shot manner. To address these limitations, we present **SketchFlow**, a novel generative framework rooted in Optimal Transport (OT) theory and flow matching. By leveraging pre-trained CLIP models to bypass labor-intensive image-level text annotations, we formulate cross-modal alignment as a continuous mapping problem directly within the CLIP latent space. To bridge the inevitable modality gap between discrete text concepts and continuous sketch features, we first inject noise into discrete category embeddings to construct a continuous Gaussian Mixture Model (GMM) prior. We then utilize an Optimal Transport Conditional Flow Matching (OT-CFM) model to learn a deterministic vector field mapping from this continuous GMM prior to the target sketch feature distribution. Finally, a Hybrid Diffusion Decoder, fusing 1D U-Net and Transformer architectures, is designed to decode these features into fast and high-fidelity stroke trajectories. Extensive experiments demonstrate that **SketchFlow** substantially outperforms existing baselines in visual quality and adherence to natural human drawing styles. Furthermore, our geometry-preserving framework demonstrates promising local zero-shot synthesis for prompts beyond the QuickDraw training vocabulary, including unseen concept labels and semantic modifiers, while enabling smooth, continuous semantic interpolation between distinct concepts. Source code is available at: https://github.com/doudin404/SketchFlow.

CCS Concepts: • **Computing methodologies** → **Shape modeling**; *Neural networks*.

Additional Key Words and Phrases: vector sketch generation, zero-shot generation, flow matching, CLIP, generative models

ACM Reference Format:
Jin Zhou, Hongliang Yang, Pengfei Xu, and Hui Huang. 2026. **SketchFlow**: Zero-Shot Vector Sketch Generation via GMM Prior Flow in CLIP Latent Space. In *SIGGRAPH Asia 2026 Conference Papers (SA Conference Papers '26), December 01–04, 2026, Kuala Lumpur, Malaysia.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3829340.3842307

1 Introduction

Vector sketches are a concise, direct medium for abstract human expression; text-conditioned vector sketch synthesis therefore has significant potential for digital avatar creation, conceptual design, and interactive art. Yet generating high-quality vector strokes with human-like drawing styles remains an open challenge. Optimization-based approaches [Arar et al. 2025; Polaczek et al. 2025; Xing et al. 2023] are slow, unstable, and prone to noise. Raster-based methods [Hu et al. 2024] lack stroke-topology constraints and introduce artifacts such as inconsistent thickness, semi-transparency, and unintended branching that impede vector tracing. Direct sequence generation with Large Language Models (LLMs) [Vinker et al. 2025]

*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.

SA Conference Papers '26, Kuala Lumpur, Malaysia

ACM ISBN 979-8-4007-2842-6/2026/12
https://doi.org/10.1145/3829340.3842307

bypasses raster-image generation but produces trajectories that deviate significantly from authentic human drawing habits in style and abstract logic.

The most intuitive approach to this task is end-to-end training using large-scale, paired text-vector sketch data. However, high-quality annotated data of this nature is exceedingly scarce. Although QuickDraw [Ha and Eck 2018], currently the largest vector sketch dataset, contains a massive collection of human-drawn trajectories, it provides only 345 discrete category labels and lacks fine-grained textual descriptions.

Consequently, we confront a fundamental research gap: How can we achieve zero-shot text-to-sketch synthesis for visual concepts absent from a fixed training vocabulary using only sketches with discrete category labels? To address this, we present **SketchFlow**, a novel generative framework rooted in Optimal Transport theory and flow matching, which constructs continuous, deterministic trajectories between distributions. By formulating cross-modal alignment as a continuous mapping problem, **SketchFlow** enables zero-shot vector sketch generation via a GMM prior flow embedded directly in the CLIP latent space. Here, *zero-shot* denotes generation from prompts whose target concept labels are absent from the 345-class QuickDraw training vocabulary, leveraging the semantic structure encoded by the pre-trained CLIP model.

Our core insight is to use pre-trained CLIP to bypass image-level text annotation and align text with sketches in a shared conceptual space. We formulate this task as a cross-modal latent space mapping problem. Raw text embeddings remain mismatched with rendered-sketch embeddings, so directly conditioning the decoder on them often fails (Figure 7). We therefore expand discrete category embeddings into a continuous GMM prior, providing a smooth source distribution for flow matching, and learn an OT-CFM vector field that transports this prior to the sketch-feature distribution. A hybrid diffusion decoder combining a 1D sequence-adapted U-Net and Transformer then maps the transported features to high-fidelity stroke trajectories.

Extensive experimental results demonstrate the superiority of our approach. This work provides a practical framework for vector sketch generation and highlights the potential of flow matching techniques in cross-modal zero-shot generation. Our core contributions are summarized as follows:

- **CLIP-mediated local zero-shot synthesis:** Our model generates recognizable sketches for unseen concept labels and semantic modifiers beyond the QuickDraw training vocabulary. The results demonstrate promising local generalization through the semantic structure encoded in the pre-trained CLIP latent space.
- **Novel cross-modal mapping framework:** We introduce an Optimal Transport Conditional Flow Matching model combined with a continuous GMM prior to effectively bridge the modality gap between discrete text concepts and continuous sketch features in the CLIP latent space.
- **High-fidelity stroke reconstruction:** We design a hybrid diffusion decoder fusing 1D U-Net and Transformer architectures, which outperforms existing baselines in visual quality and adherence to human drawing styles.

2 Related Work

2.1 Vector Sketch Generation

Vector sketches possess an inherent temporal structure; thus, early research naturally formulated the generation process as trajectory prediction over ordered strokes. SketchRNN [Ha and Eck 2018] pioneered this approach by introducing the QuickDraw dataset alongside an LSTM-VAE architecture for sketch encoding and decoding. Subsequent works [Aksan et al. 2020; Ashcroft et al. 2023; Bandyopadhyay et al. 2024; Das et al. 2021, 2023; Wang et al. 2023; Xu et al. 2026] have refined stroke modeling by utilizing RNNs, Ordinary Differential Equations (ODEs), and sequence diffusion. While these models effectively capture temporal dynamics, they typically support only unconditional or few-shot conditional generation, which severely limits their scalability and text controllability. Although Scale-Adaptive Diffusion [Hu et al. 2024] demonstrated large-scale training capabilities on the QuickDraw dataset, it generates raster images rather than scalable vector sketches. More recently, StrokeFusion [Zhou et al. 2026] proposed training a diffusion model to denoise and generate stroke sets; however, it requires a manually specified upper bound for the number of strokes and struggles to generalize to unseen categories in a zero-shot manner.

2.2 Text-Conditioned Visual Synthesis

Controllable visual synthesis accepts intuitive inputs such as text, sketches, and strokes [Bao et al. 2025; Ma et al. 2024; Xue et al. 2022]. CLIPDraw, CLIPasso, and CLIPascene optimize Bézier curves through differentiable rendering to maximize text-image similarity [Frans et al. 2022; Li et al. 2020; Vinker et al. 2023, 2022]. Without ground-truth sketch references, these methods do not reproduce authentic hand-drawn style, and direct CLIP optimization is unstable.

Diffusion-prior approaches include DiffSketcher (latent diffusion and SDS), SVGDreamer and SVGDreamer++ (vector-particle optimization), SwiftSketch (stroke-order constraints), style customization, and NeuralSVG (implicit curves) [Arar et al. 2025; Polaczek et al. 2025; Poole et al. 2022; Xing et al. 2023, 2025b, 2024; Zhang et al. 2025]. Diffusion priors also support sketch extraction [Yun et al. 2026]. These methods optimize predefined vector primitives, an initialization ill-suited to sparse hand-drawn sketches with few smooth, long strokes; SketchFlow instead learns stroke structure from human trajectories.

Language-model-based vector generation includes off-the-shelf models and models trained or fine-tuned on paired data. SketchAgent [Vinker et al. 2025] uses an off-the-shelf multimodal LLM: it provides broad semantics and logical drawing order but does not learn human drawing conventions, often producing rigid SVG graphics. IconShop, StarVector, OmniSVG, and LLM4SVG [Rodriguez et al. 2025; Wu et al. 2023; Xing et al. 2025a; Yang et al. 2025] are trained or fine-tuned on paired condition–SVG data, making generalization depend on training-condition coverage. SketchFlow instead learns human drawing behavior from category-labeled sketches and uses CLIP-space transport to generalize beyond the training vocabulary without diverse text descriptions.

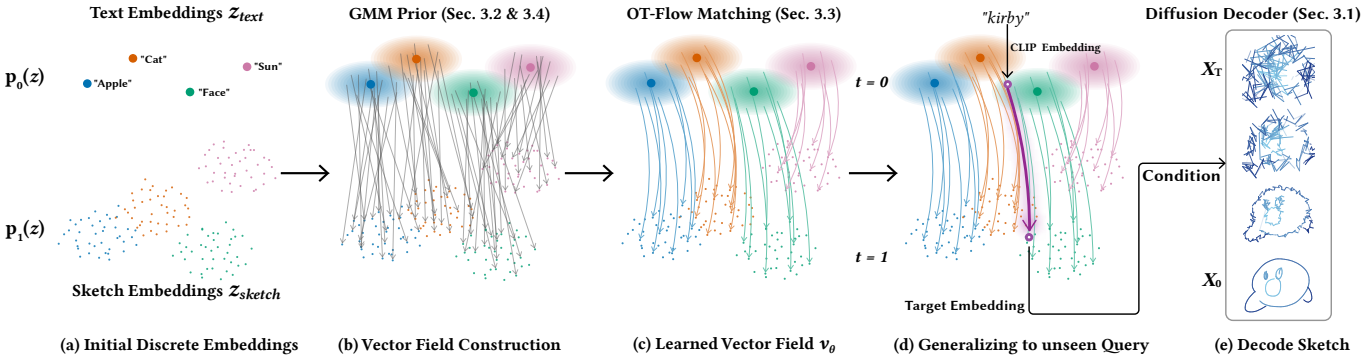

Fig. 2. **Overview of the SketchFlow generative framework.** (a) **Initial Discrete Embeddings:** The framework extracts discrete text embeddings z^k_{text} and sketch embeddings for known categories in the latent space. (b) **Vector Field Construction:** A noise injection mechanism is introduced to transform sparse concept anchors into a continuous probability density field, establishing a Gaussian Mixture Model (GMM) prior (Sec. 3.2 & 3.4). (c) **Learned Vector Field v_θ:** Optimal Transport Conditional Flow Matching establishes a deterministic mapping between the GMM prior $p_0(z)$ and the target sketch distribution, naturally yielding straight-path trajectories (Sec. 3.3). (d) **Generalizing to an Unseen Query:** During inference, the CLIP embedding of an unseen query (e.g., "kirby") follows the learned transport field, informed by known training concepts, toward the rendered-sketch CLIP feature distribution. (e) **Decode Sketch:** Conditioned on the acquired target sketch embedding, a Hybrid Diffusion Decoder utilizes a reverse diffusion process to decode pure noise X_T into a final high-fidelity vector stroke sequence X_0 (Sec. 3.1).

2.3 Flow Matching and Optimal Transport

Flow Matching [Lipman et al. 2022] has recently emerged as a powerful alternative to diffusion models. By introducing the Conditional Flow Matching (CFM) framework, it has demonstrated superior performance and stability in deterministic generation tasks. Recent advancements have deeply explored its Optimal Transport (OT) properties; methods such as Rectified Flow [Liu et al. 2022] and Optimal Transport Conditional Flow Matching (OT-CFM) [Tong et al. 2023] significantly enhance sampling efficiency by linearizing probability paths.

Recent advancements have addressed the limitations of standard Gaussian priors by employing Gaussian Mixture Models (GMMs) as Conditional Prior Distributions (CPD) [Issachar et al. 2025]. By minimizing the optimal transport cost, GMM priors substantially improve the modeling of complex data distributions. Building upon these insights, we pioneer the application of GMM-based flow matching to zero-shot vector sketch generation. Specifically, we embed a continuous GMM prior within the CLIP latent space. This approach effectively translates isolated, discrete textual categories into a continuous semantic distribution, thereby bridging the cross-modal gap between textual concepts and continuous visual trajectories.

3 Method

3.1 Sketch Representation and Hybrid Diffusion Decoder

To better understand the required inputs for sketch synthesis, we first introduce the final generation module: the Hybrid Diffusion Decoder. As illustrated in the overall framework (Figure 2), to decode target CLIP embeddings into vector stroke sequences, we design a conditional feature decoder. We represent the sketch trajectory as an equidistant sequence $S = \{(x_i, y_i, p_i)\}_{i=1}^{L}$ of L points. Here, (x_i, y_i) represents the 2D spatial coordinates. To facilitate a continuous diffusion process and avoid early over-commitment to discrete states at high-noise stages, we treat the pen-state indicator as a continuous variable $p_i \in \mathbb{R}$, scaled to a small variance range (e.g., $\{-0.1, 0.1\}$) rather than strictly binary. This ensures that p_i specifies connectivity to the preceding point without interfering with the synthesis of the global geometric trajectory during the early generation steps.

For the generative backbone, we employ a Hybrid Diffusion Decoder (Figure 3) featuring an interleaved design of 1D U-Net convolutional blocks [Ronneberger et al. 2015] and Transformer layers [Vaswani et al. 2017]. Within this architecture, the U-Net-style operations handle local geometric feature extraction and multi-resolution downsampling and upsampling, while the interleaved Transformer blocks process long-range global dependencies at their respective resolution scales.

For the conditioning mechanism, the diffusion timestep t and the target rendered-sketch CLIP embedding (denoted as z_{target}) are independently processed via sinusoidal positional encodings and linear projections. These representations are then summed to form a unified condition vector, which is injected into the hidden features of each U-Net and Transformer block using Adaptive Layer Normalization (AdaLN). The decoder is trained using a standard denoising score matching objective [Ho et al. 2020; Song et al. 2020]:

$$\mathcal{L}_{\text{Diff}} = \mathbb{E}_{x_0,\epsilon,t}\left[\left\|\epsilon - \epsilon_\theta(x_t, t, z_{\text{target}})\right\|^2\right]$$

We observe that, by using continuous CLIP embeddings rather than category labels as its semantic conditioning signal, the decoder can translate the visual semantics encoded in a rendered-sketch embedding into coherent strokes, including for unseen concepts. Therefore, the primary objective of our framework becomes acquiring the target sketch CLIP embedding directly from a textual prompt. We address this modality gap by constructing a continuous prior in the CLIP latent space, as detailed in the following section.

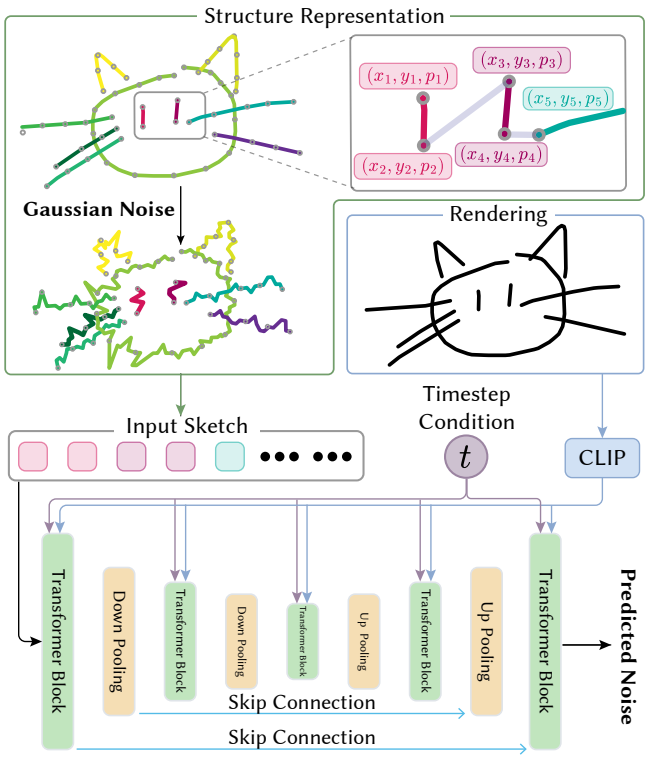

Fig. 3. **Architecture of the Hybrid Diffusion Decoder.** The model employs an interleaved generative backbone combining U-Net-style bottlenecks for local geometric extraction with Transformer blocks for global dependency modeling. Condition embeddings (diffusion timestep t and target CLIP embedding) are injected via AdaLN to predict the added noise.

3.2 GMM Prior Construction in CLIP Latent Space

To address the scarcity of fine-grained text-sketch pairs, we formulate the cross-modal alignment as a distribution transport problem in the CLIP latent space. Given that available datasets predominantly provide categorical labels rather than rich textual descriptions, we leverage the pre-trained CLIP text encoder [Radford et al. 2021] to extract discrete text embeddings z^k_{text} for each category k.

However, treating these isolated text embeddings as direct initial states does not provide continuous neighborhoods in which prompts outside the training vocabulary can be transported. To bridge this modality gap, we introduce a noise injection mechanism to construct a continuous Gaussian Mixture Model (GMM) prior. At initial state $t = 0$, we define the prior distribution $p_0(z)$ as:

$$p_0(z) = \frac{1}{K} \sum_{k=1}^{K} \mathcal{N}(z; z^k_{\text{text}}, \sigma^2 I)$$

This formulation transforms sparse concept anchors into a continuous probability density field, establishing a smooth semantic manifold from which our transport mapping originates. Here, σ represents the standard deviation of the injected noise. We treat σ as an adaptive parameter to dynamically modulate the prior, as elaborated in Sec. 3.4.

3.3 Cross-Modal Alignment via OT-Flow Matching

Given the constructed GMM prior $p_0(z)$ and the target sketch empirical distribution $p_1(z)$, our goal is to establish a deterministic mapping between these two modalities.

Conditional paradigms like unCLIP [Ramesh et al. 2022] assume a standard normal prior $p_0(z) = \mathcal{N}(0, I)$ and treat the text embedding as a condition. However, when trained on data limited to discrete categorical clusters, such formulations struggle to generalize. Ambiguous queries often cause the model to output a probabilistic mixture of isolated distributions, meaning it stochastically snaps to known modes rather than achieving true semantic interpolation.

To address this limitation, we directly formulate the continuous GMM as the base prior $p_0(z)$ and utilize Optimal Transport (OT) Conditional Flow Matching [Lipman et al. 2022; Tong et al. 2023]. The generative process follows the ODE $\mathrm{d}z_t = v_t(z_t)\mathrm{d}t$. Crucially, rather than learning an arbitrary flow, we leverage an OT coupling between p_0 and p_1 to construct the target vector field. In practice, to maintain semantic consistency during training, we construct the coupling conditionally based on semantic categories. Specifically, given a category k, we independently sample the initial state $z_0 \sim \mathcal{N}(z^k_{\text{text}}, \sigma^2 I)$ (σ is the adaptive prior variance detailed in Sec. 3.4) and the target state $z_1 \sim p_1(z|k)$. The flow matching model is then trained to regress the optimal transport vector field, which trivially forms a straight path $z_0 \to z_1$, using the Conditional Flow Matching objective [Lipman et al. 2022]:

$$\mathcal{L}_{\text{CFM}} = \mathbb{E}_{t, p_0(z_0|k), p_1(z_1|k)} \left[\| v_\theta(z_t, t, c, \sigma) - (z_1 - z_0) \|^2 \right]$$

where the intermediate state is parameterized as $z_t = (1-t)z_0 + tz_1$.

This OT formulation minimizes transport cost and yields straight-path, constant-velocity trajectories with two useful properties. (1) **Semantic Direction Preservation:** Straight paths preserve semantic feature directions in the CLIP latent space. (2) **Non-Crossing Sampling Paths:** The learned ODE flow produces non-intersecting paths that preserve the relative organization of semantic regions, allowing unseen queries to follow the surrounding transport toward semantically related regions of the rendered-sketch CLIP space.

3.4 Adaptive Prior Variance Injection

Empirical results suggest that a single fixed noise scale σ is suboptimal for constructing the initial GMM prior across all semantic categories. Different semantic concepts intrinsically require varying degrees of spatial dispersion to achieve optimal generation fidelity.

To address this, we introduce a tunable prior variance mechanism. We condition the flow-matching vector field on the prior variance as $v_\theta(z_t, t, c, \sigma)$. Specifically, to construct this conditioning mechanism, the flow timestep t, the text embedding c, and the logarithm of the prior variance $\log(\sigma)$ are independently processed via sinusoidal positional encodings and linear projections. These representations are then summed to form a unified condition vector, which is injected into the hidden features of the flow matching network using Feature-wise Linear Modulation (FiLM) [Perez et al. 2018].

This allows the generated marginal distribution at the endpoint to be dynamically modulated, denoted as $\hat{p}_1(z \mid z_{\text{test}}, \sigma)$. During training, we randomly sample standard deviations σ to modulate the initial GMM prior, directly shaping the generative trajectory.

During inference, this formulation allows users to adaptively adjust σ based on specific prompts. Specifically, we introduce a scaling parameter γ to control the sampling standard deviation of the query point relative to the standard deviation of the input condition.

4 Experiments

4.1 Experimental Setup

Datasets. We use all 345 categories in the QuickDraw release distributed for Sketch-RNN. Each category contains 70,000 training, 2,500 validation, and 2,500 test sketches, yielding 24.15 million, 862,500, and 862,500 samples in the respective splits. We extract stroke trajectories for sequence-level modeling. Furthermore, we conducted experiments on cross-domain transfer datasets to evaluate the model's generalizability; these extensive results are provided in the supplementary material.

Implementation Details. For data preprocessing, we resample sketch sequences to $L = 256$ and scale binary pen-state indicators to $\{-0.1, 0.1\}$ to prevent premature commitment to discrete states during high-noise diffusion stages. During training, sketches are rendered in black and white for CLIP feature extraction via the pre-trained ViT-B-32. For visual presentation in this paper, generated stroke trajectories are explicitly color-coded from light to dark to illustrate the sequential drawing order. The OT-Flow Matching model is parameterized by a residual MLP with FiLM layers, while the Hybrid Decoder utilizes a 1D U-Net interleaved with Transformer blocks. The transport model and decoder are jointly optimized for 1,221,444 steps (518 epochs, using 20% of the shuffled training loader per epoch) with AdamW and a total batch size of 256. Training takes approximately three days on eight NVIDIA RTX 3090 GPUs. At inference, we use 60 steps for both components: Heun's method integrates the transport ODE, followed by 60 DDPM denoising steps in the decoder. Comprehensive architectural and training hyperparameters are provided in the supplementary material.

Evaluation Metrics. Visual fidelity is evaluated using FID on 256×256 rasterized renderings. Cross-modal semantic alignment is measured by CLIP Score using the prompt "a sketch of [category]". To assess trajectory abstraction, we evaluate the *RDP Point Count* by computing the absolute number of remaining anchor points after applying the Ramer-Douglas-Peucker algorithm [Ramer 1972]. Before simplification, each trajectory is isotropically normalized so that the longest side of its bounding box is 1,000; we then use a tolerance of $\epsilon = 2.0$. Fewer remaining points indicate higher abstraction, avoiding biases from the original sketch resolution.

Baselines. We compare representative generative paradigms: (1) **SketchRNN** [Ha and Eck 2018], an LSTM-VAE trajectory model; (2) **StrokeFusion** [Zhou et al. 2026], a latent trajectory diffusion model; (3) **NeuralSVG** [Polaczek et al. 2025], an SDS-based optimization approach; and (4) **SketchAgent** [Vinker et al. 2025], an LLM-driven planning agent. To ensure fair comparison, we omit FID scores for optimization-based and LLM-driven methods, as they are not explicitly trained on our data distribution. For complementary open-prompt comparison, we additionally evaluate a two-stage pipeline combining T2I and CLIPasso [Vinker et al. 2022] on matched unseen prompts; the complete qualitative comparison is provided in the supplementary material.

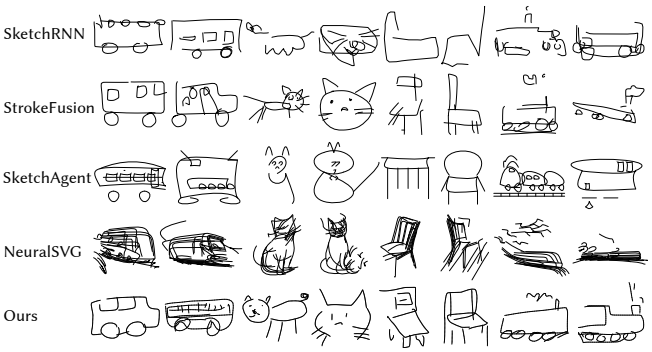

Fig. 4. **Qualitative comparison on QuickDraw.** Generated samples for four categories (*Bus*, *Cat*, *Chair*, and *Train*) from baseline methods and ours.

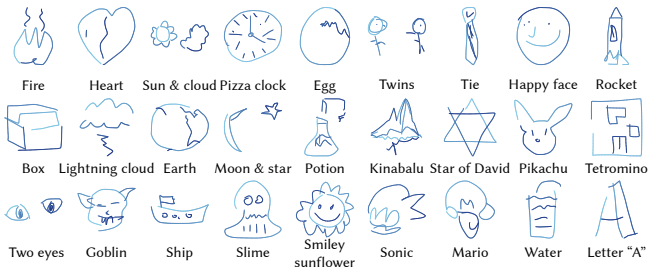

Fig. 5. **Zero-shot generation beyond the training vocabulary.** All prompts shown here lie outside the QuickDraw taxonomy, yet the model produces coherent and style-consistent vector sketches.

4.2 Comparison with Existing Methods

To evaluate our approach against existing state-of-the-art methods, we select four representative categories: *Bus*, *Cat*, *Chair*, and *Train*. These categories exhibit relatively complex structural characteristics and are included in the known classes of the QuickDraw dataset, facilitating a fair and direct in-domain generation-quality comparison with other QuickDraw-based methods.

For each category, we generate 10,000 samples and compute the three evaluation metrics introduced in Sec. 4. To generate a specific category, the category name is used as the conditioning text input. Additional stochasticity is introduced by adding Gaussian noise scaled by the factor γ, as defined in Sec. 3.4. We report our results under varying noise scales $\gamma \in \{0.6, 0.8, 1.0\}$, while the base prior variance σ is fixed at 0.025. The corresponding quantitative evaluations and qualitative visualizations are presented in Table 1 and Figure 4, respectively.

As shown in Table 1, our method achieves significant improvements in the FID metric compared to all learning-based baselines across all categories. This substantial margin demonstrates that our model possesses superior representation and feature learning capabilities. Although some baseline methods achieve lower RDP scores, this is primarily because they tend to learn and generate

Fig. 6. **Zero-Shot Generation with Semantic Modifiers.** Generation results for specific action prompts. Three distinct results are shown for each prompt.

overly simplistic or rudimentary sketches, which is demonstrated by the qualitative results in Figure 4.

Furthermore, our generated results achieve CLIP scores comparable to those of NeuralSVG and SketchAgent. However, the higher RDP scores of these baseline methods suggest a larger number of points after simplification, which often corresponds to cluttered and unstructured strokes. While such visual complexity may inadvertently inflate CLIP recognition scores, it deviates from actual human drawing behaviors. In contrast, qualitative comparisons in Figure 4 illustrate that **SketchFlow** better captures the abstract characteristics of hand-drawn sketches, producing trajectories that are concise, coherent, and effective at conveying target concepts.

4.3 Zero-Shot Generation Beyond the Training Vocabulary

Beyond generating known classes from the training dataset, our model demonstrates zero-shot generalization to labels outside the 345-class QuickDraw training vocabulary through OT-Flow Matching in the CLIP latent space. Figure 5 presents results for several unseen prompts, with additional results provided in the supplementary material. These include concepts such as *Pikachu*, *Tetromino*, and *Goblin*, as well as recognizable outputs for the composite and plural prompts *Sun & cloud* and *Two eyes*. These results demonstrate that **SketchFlow** leverages visual semantics encoded by CLIP to generate vector sketches for unseen concepts.

To quantify generalization beyond selected examples, we evaluate all 44 prompts whose complete labels are absent from the 345-class training vocabulary. For each prompt, we generate 32 samples and report their mean CLIP Score as a label-alignment measure, without sample selection. The Interp CLIP baseline fits affine weights over the top-8 nearest QuickDraw text anchors, allowing extrapolation, and applies the weights to the corresponding sketch-latent anchors. As shown in Table 2, **SketchFlow** achieves 0.2480, compared with 0.2024 for the Gaussian-Prior baseline and 0.2034 for Interp CLIP, outperforming Interp CLIP on all 44 prompts. Complete per-prompt results are provided in the supplementary material.

Figure 6 further shows that the model responds to semantic modifiers and actions even though it was trained exclusively on discrete category names. For prompts such as *a running cat*, *a jumping cat*, and *a stretching cat*, the generated trajectories reflect distinct dynamic poses while maintaining the abstract hand-drawn style. This provides qualitative evidence that the learned CLIP-space transport retains modifier semantics beyond the discrete category names used for training.

4.4 User Studies

Following [Bhunia et al. 2022; Wang et al. 2025], we conduct a user study of perceptual quality and human-likeness, comparing **SketchFlow** with NeuralSVG [Polaczek et al. 2025] and SketchAgent [Vinker et al. 2025]. To account for the variance in generation quality among these methods and ensure a fair evaluation, we selected the single best result (highest CLIP score) from 10 generated samples per prompt. A total of 42 participants were recruited for this study. They assessed the sketches across five metrics: abstraction fidelity, semantic alignment, human-likeness, stroke rationality, and aesthetic quality. As shown in Table 3, **SketchFlow** consistently outperforms the comparison methods across all criteria. This strong preference demonstrates that our approach successfully captures the abstract semantic essence of concepts and naturally simulates authentic human drawing nuances (*e.g.*, slight stroke variations).

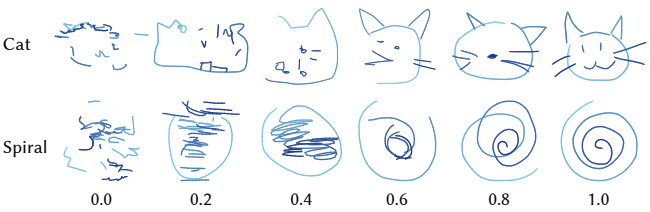

Fig. 7. **Ablation of Flow Mapping.** Results of linear interpolating the condition from raw text embeddings ($\lambda = 0.0$) to Flow-mapped embeddings ($\lambda = 1.0$) at intervals of 0.2. The raw text embedding alone is insufficient to guide structured sketch synthesis.

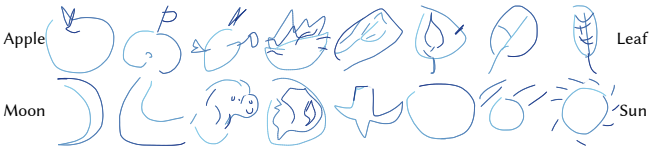

Fig. 8. **Ablation of Gaussian-Prior baseline** Latent interpolation results produced by the baseline. Unlike our approach, this baseline struggles to maintain structural coherence during the transition. The intermediate sketches exhibit abrupt topological shifts and a loss of visual semantics, highlighting the necessity of our GMM-based OT-Flow Matching design for establishing a continuous generative space.

4.5 Continuous Interpolation

To demonstrate that our framework learns a continuous semantic manifold rather than merely overfitting to discrete training categories, we perform linear interpolation between distinct text embeddings. As illustrated in Figure 10, the model generates smooth and topologically coherent structural transitions between different concepts, validating the continuity of the established generative space.

4.6 Ablation Studies

CLIP text and sketch embeddings follow misaligned distributions; our OT-Flow Matching module bridges this cross-modal gap. Figure 7 ablates flow mapping by linearly interpolating the condition from the raw CLIP text embedding ($\lambda = 0.0$) to its flow-mapped target ($\lambda = 1.0$). Raw text conditioning misses the valid sketch manifold and produces unstructured, unrecognizable strokes. As

Table 1. Quantitative results on QuickDraw *Bus*, *Cat*, *Chair*, and *Train*. Bold and underlined values indicate the best and second-best performances, respectively.

	Bus			*Cat*			*Chair*			*Train*		
Method	FID ↓	CLIP ↑	RDP ↓	FID ↓	CLIP ↑	RDP ↓	FID ↓	CLIP ↑	RDP ↓	FID ↓	CLIP ↑	RDP ↓
SketchRNN	66.103	0.266	**79.571**	70.704	0.210	**68.591**	88.294	0.227	<u>25.351</u>	83.578	0.236	**78.830**
StrokeFusion	59.329	0.278	86.690	52.940	0.260	85.730	33.463	0.277	36.460	59.418	0.247	102.910
SketchAgent	–	0.258	124.429	–	0.214	93.684	–	0.267	40.200	–	0.234	98.400
NeuralSVG	–	0.224	278.446	–	<u>0.261</u>	267.091	–	**0.304**	225.663	–	0.225	256.168
Gaussian-Prior	9.556	0.265	98.53	24.568	0.148	97.15	17.572	0.165	38.45	15.180	0.183	99.06
Ours ($\gamma = 0.60$)	19.381	**0.287**	<u>85.47</u>	27.977	**0.262**	<u>79.77</u>	41.123	<u>0.294</u>	**24.30**	15.347	**0.260**	<u>92.29</u>
Ours ($\gamma = 0.80$)	<u>9.004</u>	<u>0.284</u>	92.12	<u>17.899</u>	0.261	86.33	<u>21.545</u>	0.287	29.75	<u>8.217</u>	<u>0.256</u>	96.21
Ours ($\gamma = 1.00$)	**5.810**	0.276	100.35	**14.423**	0.254	95.75	**11.718**	0.280	36.65	**7.181**	0.248	103.79
Dataset	0	0.284	91.88	0	0.259	77.83	0	0.283	29.29	0	0.266	101.89

Table 2. Mean CLIP Score across 44 prompts absent from the QuickDraw training vocabulary. Each value averages 32 samples per prompt without best-of-N selection ($\gamma = 1.0$).

Method	Mean CLIP Score ↑
Gaussian Prior	0.2024
Interp CLIP	0.2034
SketchFlow (Ours)	**0.2480**

Table 3. User study results comparing **SketchFlow** with **NeuralSVG** and **SketchAgent** across five metrics. The scores indicate the average percentage of user preference for each method.

Metric	**SketchFlow**	**NeuralSVG**	**SketchAgent**
Abstraction Fidelity	72.85%	12.52%	14.63%
Semantic Alignment	56.75%	32.52%	10.73%
Human-likeness	70.73%	16.59%	12.68%
Stroke Rationality	61.79%	26.18%	12.03%
Aesthetic Quality	51.71%	37.24%	11.05%

(a) Incomplete structures **(b) Text-like strokes**

Fig. 9. **Representative failure cases.** (a) The outputs remain recognizable as the prompted concepts but contain incomplete structures. (b) The generated trajectories form text-like strokes resembling the prompt words instead of the intended objects.

the condition approaches the flow-mapped output, the trajectories smoothly self-organize into structurally coherent, recognizable sketches. To empirically validate the necessity of our GMM prior formulation over the conventional conditional paradigm (Sec. 3.3), we train a variant denoted as the "Gaussian-Prior Baseline." This model follows the unCLIP approach by assuming a standard normal prior $p_0(z) = \mathcal{N}(0, I)$ and treating the text embedding strictly as a condition.

The quantitative evaluation of this baseline is recorded in Table 1, where it exhibits a clear performance degradation compared to our method. This decline stems from the unCLIP paradigm's inability to explicitly align or organize the cross-modal distributions. Consequently, a perturbed conditional embedding from one category can easily drift into the latent representation of another, resulting in a noticeable probability of generating sketches that deviate from the input class.

More importantly, as discussed in Section 3.3, this purely conditional model generalizes poorly to the unseen prompts considered in our experiments. To illustrate this, we replicate the latent interpolation experiment using the unCLIP baseline, with results shown in Figure 8 and Figure 10. The generated sketches exhibit abrupt structural shifts and rapidly lose visual readability. Rather than traversing a continuous semantic manifold, the model simply snaps between isolated training modes, failing to synthesize structurally coherent intermediate concepts.

4.7 Image-Conditioned Generation

Because CLIP aligns vision and language in latent space, replacing the text prompt with a reference-image embedding extends **SketchFlow** to zero-shot semantic abstraction and style transfer without architectural changes or paired image-sketch fine-tuning. As shown in Figure 11, **SketchFlow** captures the input's core semantics, geometry, and pose in abstraction style.

5 Conclusion & Limitations

Conclusion. We present **SketchFlow**, a zero-shot text-to-vector sketch framework that bridges discrete text concepts and continuous sketch features through OT-CFM and a continuous GMM prior in CLIP space. Its fast forward-pass generator achieves high visual quality, follows human drawing styles, and shows promising local zero-shot generalization beyond the QuickDraw vocabulary.

The results demonstrate that Optimal Transport geometry enables CLIP-mediated zero-shot transport under label-only textual supervision, reducing reliance on exhaustive fine-grained textual annotations for individual training sketches.

Limitations. The text-to-sketch mapping remains imperfect. As shown in Figure 9, some outputs for prompts beyond the training vocabulary retain recognizable concept-level structure but omit parts of the intended shape, while others drift toward text-like strokes resembling the prompt word rather than depicting the intended object. Moreover, some unseen categories require increased search variance (γ) and still yield inconsistent results. We hypothesize that this limitation reflects the locally uniform Gaussian assumption around text anchors, whereas semantic manifolds can be complex and non-Gaussian. Future work will develop more precise latent mappings to better capture these non-Gaussian semantic structures.

Acknowledgments

We thank the anonymous reviewers for their constructive feedback and the user study participants for their time and effort. This work was supported in part by National Natural Science Foundation of China (62472287), Natural Science Foundation of Shenzhen City (JCYJ20250604181519025), and Scientific Development Fund from Guangdong Provincial Key Laboratory of Visual Media and Multidimensional Intelligence.

References

Emre Aksan, Thomas Deselaers, Andrea Tagliasacchi, and Otmar Hilliges. 2020. Cose: Compositional stroke embeddings. In *NeurIPS*.

Ellie Arar, Yarden Frenkel, Daniel Cohen-Or, Ariel Shamir, and Yael Vinker. 2025. Swiftsketch: A diffusion model for image-to-vector sketch generation. In *SIGGRAPH*. 1–12.

Alexander Ashcroft, Ayan Das, Yulia Gryaditskaya, Zhiyu Qu, and Yi-Zhe Song. 2023. Modelling complex vector drawings with stroke-clouds. In *ICLR*.

Hmrishav Bandyopadhyay, Ayan Kumar Bhunia, Pinaki Nath Chowdhury, Aneeshan Sain, Tao Xiang, Timothy Hospedales, and Yi-Zhe Song. 2024. Sketchinr: A first look into sketches as implicit neural representations. In *CVPR*. 12565–12574.

Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. 2025. Text to Image Generation with Bidirectional Multiway Transformers. *Computational Visual Media* 11, 2 (2025), 405–422.

Ankan Kumar Bhunia, Salman Khan, Hisham Cholakkal, Rao Muhammad Anwer, Fahad Shahbaz Khan, Jorma Laaksonen, and Michael Felsberg. 2022. Doodleformer: Creative sketch drawing with transformers. In *ECCV*. Springer, 338–355.

Ayan Das, Yongxin Yang, Timothy Hospedales, Tao Xiang, and Yi-Zhe Song. 2021. Sketchode: Learning neural sketch representation in continuous time. In *ICLR*.

Ayan Das, Yongxin Yang, Timothy Hospedales, Tao Xiang, and Yi-Zhe Song. 2023. Chirodiff: Modelling chirographic data with diffusion models. In *ICLR*.

Kevin Frans, Lisa Soros, and Olaf Witkowski. 2022. Clipdraw: Exploring text-to-drawing synthesis through language-image encoders. In *NeurIPS*.

David Ha and Douglas Eck. 2018. A neural representation of sketch drawings. In *ICLR*.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. In *NeurIPS*.

Jijin Hu, Ke Li, Yonggang Qi, and Yi-Zhe Song. 2024. Scale-adaptive diffusion model for complex sketch synthesis. In *ICLR*.

Noam Issachar, Mohammad Salama, Raanan Fattal, and Sagie Benaim. 2025. Designing a Conditional Prior Distribution for Flow-Based Generative Models. *TMLR* (2025).

Tzu-Mao Li, Michal Lukáč, Michaël Gharbi, and Jonathan Ragan-Kelley. 2020. Differentiable vector graphics rasterization for editing and learning. *ACM TOG* 39, 6 (2020), 1–15.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. 2022. Flow matching for generative modeling. In *ICLR*.

Xingchao Liu, Chengyue Gong, and Qiang Liu. 2022. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *ICLR*.

Hao Ma, Ming Li, Jingyuan Yang, Or Patashnik, Dani Lischinski, Daniel Cohen-Or, and Hui Huang. 2024. CLIP-Flow: Decoding Images Encoded in CLIP Space. *Computational Visual Media* 10, 6 (2024), 1157–1168.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. 2018. Film: Visual reasoning with a general conditioning layer. In *AAAI*, Vol. 32.

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.

Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. 2022. DreamFusion: Text-to-3D using 2D Diffusion. In *ICLR*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *ICML*. PmLR, 8748–8763.

Urs Ramer. 1972. An iterative procedure for the polygonal approximation of plane curves. *CGIP* 1, 3 (1972), 244–256.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125* 1, 2 (2022), 3.

Juan A. Rodriguez, Abhay Puri, Shubham Agarwal, Issam H. Laradji, Pau Rodriguez, Sai Rajeswar, David Vazquez, Christopher Pal, and Marco Pedersoli. 2025. StarVector: Generating Scalable Vector Graphics Code from Images and Text. In *CVPR*. 16175–16186.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*. Springer, 234–241.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. 2020. Score-based generative modeling through stochastic differential equations. In *ICLR*.

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. 2023. Improving and generalizing flow-based generative models with minibatch optimal transport. *TMLR* (2023).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*, Vol. 30.

Yael Vinker, Yuval Alaluf, Daniel Cohen-Or, and Ariel Shamir. 2023. Clipascene: Scene sketching with different types and levels of abstraction. In *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 TOG* 41, 4 (2022), 1–11.

Yael Vinker, Tamar Rott Shaham, Kristine Zheng, Alex Zhao, Judith E Fan, and Antonio Torralba. 2025. Sketchagent: Language-driven sequential sketch generation. In *CVPR*. 23355–23368.

Jiawei Wang, Zhiming Cui, and Changjian Li. 2025. VQ-SGen: A Vector Quantized Stroke Representation for Sketch Generation. In *ICCV*.

Qiang Wang, Haoge Deng, Yonggang Qi, Da Li, and Yi-Zhe Song. 2023. Sketchknitter: Vectorized sketch generation with diffusion models. In *ICLR*.

Ronghuan Wu, Wanchao Su, Kede Ma, and Jing Liao. 2023. Iconshop: Text-guided vector icon synthesis with autoregressive transformers. *ACM TOG* 42, 6 (2023), 1–14.

Ximing Xing, Juncheng Hu, Guotao Liang, Jing Zhang, Dong Xu, and Qian Yu. 2025a. Empowering LLMs to Understand and Generate Complex Vector Graphics. In *CVPR*. 19487–19497.

Ximing Xing, Chuang Wang, Haitao Zhou, Jing Zhang, Qian Yu, and Dong Xu. 2023. Diffsketcher: Text guided vector sketch synthesis through latent diffusion models. *NeurIPS* 36 (2023), 15869–15889.

Ximing Xing, Qian Yu, Chuang Wang, Haitao Zhou, Jing Zhang, and Dong Xu. 2025b. SVGDreamer++: Advancing Editability and Diversity in Text-Guided SVG Generation. *IEEE TPAMI* 47, 7 (2025), 5397–5413.

Ximing Xing, Haitao Zhou, Chuang Wang, Jing Zhang, Dong Xu, and Qian Yu. 2024. SVGDreamer: Text Guided SVG Generation with Diffusion Model. In *CVPR*. 4546–4555.

Pengfei Xu, Banhuai Ruan, Youyi Zheng, and Hui Huang. 2026. Sketchformer++: A Hierarchical Transformer Architecture for Vector Sketch Representation. *Computational Visual Media* 12, 1 (2026), 173–188.

Yuan Xue, Yuan-Chen Guo, Han Zhang, Tao Xu, Song-Hai Zhang, and Xiaolei Huang. 2022. Deep Image Synthesis from Intuitive User Input: A Review and Perspectives. *Computational Visual Media* 8, 1 (2022), 3–31.

Yiying Yang, Wei Cheng, Sijin Chen, Xianfang Zeng, Fukun Yin, Jiaxu Zhang, Liao Wang, Gang Yu, Xingjun Ma, and Yu-Gang Jiang. 2025. OmniSVG: A Unified Scalable Vector Graphics Generation Model. In *NeurIPS*, Vol. 38.

Kwan Yun, Youngseo Kim, Kwanggyoon Seo, Chang Wook Seo, and Junyong Noh. 2026. Representative Feature Extraction During Diffusion Process for Sketch Extraction with One Example. *Computational Visual Media* (2026), 1–25. Online first.

Peiying Zhang, Nanxuan Zhao, and Jing Liao. 2025. Style Customization of Text-to-Vector Generation with Image Diffusion Priors. In *SIGGRAPH*. 1–11.

Jin Zhou, Yi Zhou, Hongliang Yang, Pengfei Xu, and Hui Huang. 2026. StrokeFusion: Vector Sketch Generation via Joint Stroke-UDF Encoding and Latent Sequence Diffusion. In *AAAI*, Vol. 40. 13656–13664.

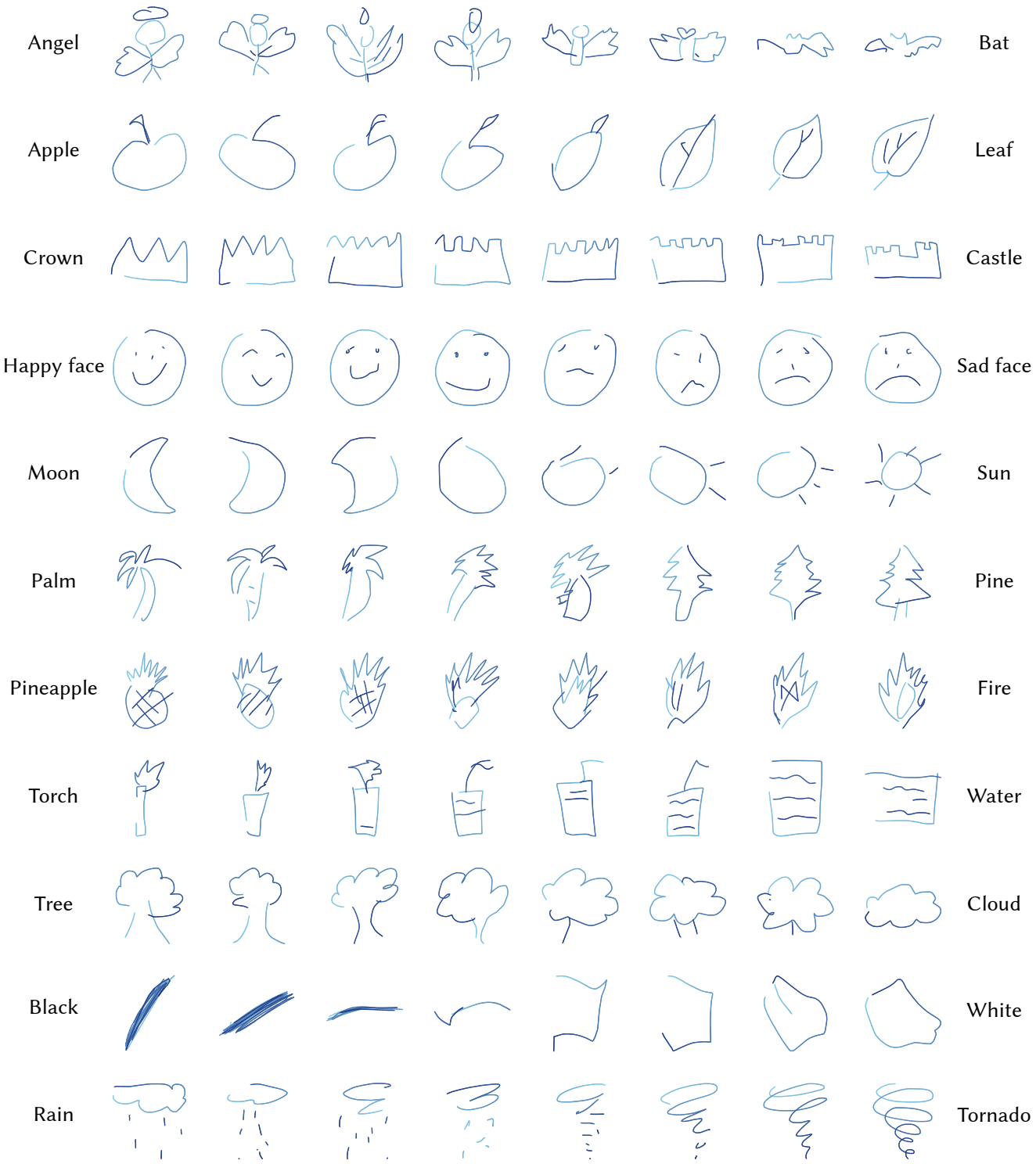

Fig. 10. **Semantic Interpolation in CLIP Space.** Each row visualizes a linear interpolation path between the discrete text embeddings of two distinct concepts (e.g., smoothly morphing from an "Apple" to a "Leaf", or from a "Palm" to a "Pine"). The intermediate generations maintain exceptionally uniform structural transitions and logical stroke topologies without disjointed visual jumps. This progressive morphing provides compelling evidence that our model successfully establishes a continuous generative manifold, rather than stochastically snapping to isolated known modes.

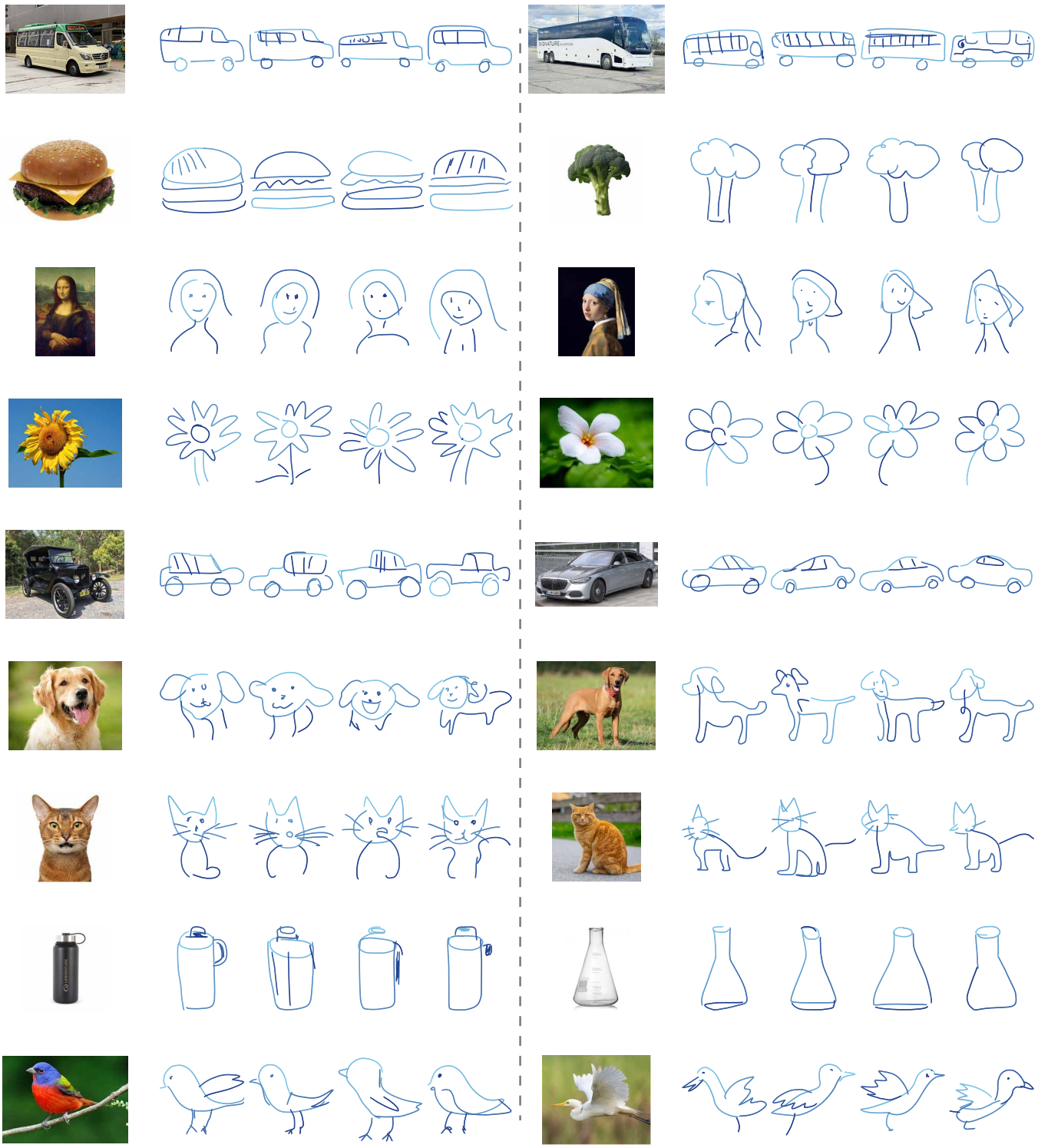

Fig. 11. **Zero-shot Image-Conditioned Vector Sketch Synthesis.** As an intriguing extension, our model can seamlessly accept CLIP visual embeddings in place of textual prompts. This substitution enables a zero-shot stylized redrawing effect, where the generated vector sketches effectively capture the core semantic features and structural characteristics of the reference images.