StrokeFusion: Vector Sketch Generation via Joint Stroke-UDF Encoding and Latent Sequence Diffusion

Jin Zhou, Yi Zhou, Hongliang Yang, Pengfei Xu*, Hui Huang

College of Computer Science and Software Engineering, Shenzhen University, China

Abstract

In the field of sketch generation, raster-format-trained models often produce non-stroke artifacts, while vector-format-trained models typically lack a holistic understanding of sketches, resulting in compromised recognizability. Moreover, existing methods struggle to extract common features from similar elements (e.g., animal eyes) that appear at varying positions across sketches. To address these challenges, we propose **StrokeFusion**, a two-stage framework for vector sketch generation. It contains a dual-modal sketch feature learning network that maps strokes into a high-quality latent space. This network decomposes sketches into normalized strokes and jointly encodes stroke sequences with Unsigned Distance Function (UDF) maps, representing sketches as sets of stroke feature vectors. Building upon this representation, our framework exploits a stroke-level latent diffusion model that simultaneously adjusts stroke position, scale, and trajectory during generation. This enables high-fidelity stroke generation while supporting stroke interpolation editing. Extensive experiments across multiple sketch datasets demonstrate that our framework outperforms state-of-the-art techniques, validating its effectiveness in preserving structural integrity and semantic features.

Code — https://github.com/doudin404/StrokeFusion

Introduction

Sketch generation, as an essential component of computational creativity, significantly accelerates concept visualization and rapid design iteration in numerous fields, including product design, animation, and interactive prototyping. Although humans effortlessly produce and interpret sketches by intuitively capturing holistic structures and local details, existing computational methods fall short of emulating this capability. These limitations manifest as difficulties in capturing global semantic structures, maintaining stroke-level control, and generating visually coherent sketches, thereby limiting their practicality in professional and creative workflows. Consequently, addressing these challenges through tailored computational paradigms is imperative.

Current sketch representations primarily exist in two formats: *raster sketches* and *vector sketches*. Raster sketches render strokes directly as images and are learned through reconstruction of raster images (Wang, Cui, and Li 2025; Ge et al. 2020; Bhunia et al. 2022), allowing the use of mature image representation learning methods. However, these models often produce pixelation artifacts and non-stroke noise. More critically, they discard stroke trajectories, making it difficult to capture human drawing habits and natural stroke patterns, focusing solely on overall shape.

In contrast, vector sketches preserve clean and precise stroke trajectories that better simulate human drawing processes. Current vector-based approaches typically trace the entire stroke trajectory as polylines, recording pen-up and pen-down states. This leads to two modeling paradigms: modeling strokes using velocity-based relative coordinates (Ha and Eck 2018; Das et al. 2023), or using absolute position representations (Wang et al. 2023; Bandyopadhyay et al. 2024). Relative coordinate modeling allows the model to learn common patterns more easily (e.g., eyes, wheels), since velocity vectors are often similar across these structures. However, these models must integrate velocity vectors to determine current positions, leading to error accumulation that makes it difficult to form closed loops or to determine position components accurately. Absolute coordinate modeling alleviates these issues but weakens the model's ability to capture stroke-wise commonalities, leading to inferior quality compared to velocity-based methods.

To address these challenges, we propose **StrokeFusion**, a two-stage framework that combines dual-modal feature encoding with latent space diffusion. In the first stage, we introduce *stroke-UDF joint encoding*, which fuses vector primitives with rasterized unsigned distance fields (UDFs) to capture both geometric structure and stroke-level semantics. In the second stage, a *stroke-level latent diffusion model* generates strokes in a non-autoregressive and order-invariant manner by jointly denoising position, scale, and trajectory in a structured latent space.

Our main contributions are:

- A dual-modal encoding scheme that integrates vector and raster features to enhance stroke representation while preserving spatial and semantic attributes.
- A disentangled design that separates stroke layout prediction and shape synthesis, improving structural consistency and generation controllability.

*Corresponding author, e-mail: xupengfei.cg@gmail.com

- A latent diffusion model that supports unordered, variable-length stroke generation, overcoming the limitations of sequential models.

Extensive experiments demonstrate that our method significantly outperforms baseline approaches by leveraging stroke-sketch hierarchical structures. Both quantitative metrics and qualitative comparisons reveal the advantages of our framework. Ablation studies on dual-modal encoding, stroke latent space, and spatial information disentanglement further validate the effectiveness of our design choices.

Related Work

Raster Sketch Generation

Various methods focus on generating raster sketches. DoodlerGAN (Ge et al. 2020) introduced a part-based GAN framework for learning the appearance of semantic parts in image space. DoodleFormer (Bhunia et al. 2022) improved structural understanding using a coarse-to-fine generation strategy with a two-stage Transformer. VQ-SGen (Wang, Cui, and Li 2025) implemented vector quantization to encode sketches into discrete stroke patches for autoregressive decoding. While these methods yield visually appealing results, they do not produce editable vector outputs. Our approach combines image-level supervision in shape modeling with the benefits of vector-format outputs.

Vector Sketch Generation

Ha et al. (Ha and Eck 2018) developed stroke-annotated polyline sequences with the Sketch-RNN model. Ribeiro et al. (Ribeiro et al. 2020), Xu et al. (Xu et al. 2024), and Lin et al. (Lin et al. 2020) enhanced Transformer-based architectures for representation learning aimed at recognition and completion rather than generative modeling. Qi et al. (Qi et al. 2022) used Graph Convolutional Networks (Kipf and Welling 2017) for local structure modeling. Tiwari et al. (Tiwari, Biswas, and Lladós 2024) tokenized sketches for GPT-style decoding, while Zang et al. (Zang, Tu, and Xu 2023) used Gaussian mixture latent spaces for structure-aware priors. Wang et al. (Wang et al. 2023) represented vector sketches as offset sequences to improve diffusion efficiency. Das et al. (Das et al. 2023) approached stroke generation as continuous functions. However, many approaches overfit to superficial details, leading to inefficient training and poor generalization. Inspired by recent progress in 3D shape generation (Zhang et al. 2023), our method encodes strokes into fixed-length feature vectors, treating them as unordered sets to handle variability in stroke count and order.

Continuous Vector Sketch Generation

Traditional methods often discretize sketches, overlooking the continuous nature of human drawing. To address this, several methods explore continuous representations. PMN (Alaniz et al. 2022) used primitives from a graphical library, while BézierSketch (Das et al. 2020) and Cloud2Curve (Das et al. 2021) employed Bézier curves with adaptive orders. Stroke Clouds (Ashcroft et al. 2023) used diffusion models for third-order Bézier curves, but these techniques suffer from limited expressiveness.

Recent work focuses on fully continuous representations. SketchODE (Das et al. 2022) applied Neural ODEs for stroke dynamics, and SketchINR (Bandyopadhyay et al. 2024) used implicit neural representations. However, these methods can miss holistic coherence. Our approach encodes entire strokes as learnable embeddings, enabling semantically coherent sketch composition while maintaining clean vector trajectories.

Diffusion Models

Our work is rooted in Diffusion Models (Sohl-Dickstein et al. 2015), which involve forward and reverse processes. Recent advances include continuous-space implementations (Dhariwal and Nichol 2021; Ho, Jain, and Abbeel 2020) and discrete-space extensions like D3PM (Austin et al. 2021). We incorporate these advancements in diffusion modeling and align them with recent progress in 3D shape representation.

Method

Problem Formulation and Representation

Existing sketch generation approaches face a fundamental representation dilemma: point-sequence models capture stroke geometry but struggle with visual fidelity, while raster-based methods preserve appearance yet lose structure-level semantics. To bridge this gap, we propose a two-stage sketch generation framework comprising (1) *stroke-UDF joint encoding*, which fuses vector stroke geometry and spatial structure, and (2) a *stroke-level latent diffusion model* that synthesizes strokes in an unordered, non-autoregressive fashion. In this section, we detail our representation strategies for both vector and raster modalities.

Vector Representation

Depending on data sources, input strokes may be represented as Bézier curves or RDP-simplified polylines. To ensure representation consistency across different formats and abstraction levels, we resample each stroke into N_p evenly spaced points along its trajectory. Thus, each stroke s_j is represented as a sequence of points:

$$s_j = p_1, p_2, \ldots, p_{N_p}, \quad p_i = (x_i, y_i), \tag{1}$$

where (x_i, y_i) are normalized coordinates obtained with respect to the stroke's tight bounding square. This normalization guarantees scale invariance and facilitates stroke-wise feature extraction.

Visual Representation

To capture stroke shape in a raster-friendly manner, we convert each stroke into an unsigned distance field $I_u(g)$ using aggregated stroke density fields (Alaniz et al. 2022). For each segment defined by two consecutive points x_i and x_{i+1}, we compute interpolated locations:

$$p_i(r) = r x_i + (1-r) x_{i+1}, \quad r \in [0, 1]. \tag{2}$$

Then, the exponential decay of distance from any grid location g to a stroke segment is defined as:

$$d_i(g) = \max_{r \in [0,1]} \exp\left(-\gamma |g - p_i(r)|^2\right), \tag{3}$$

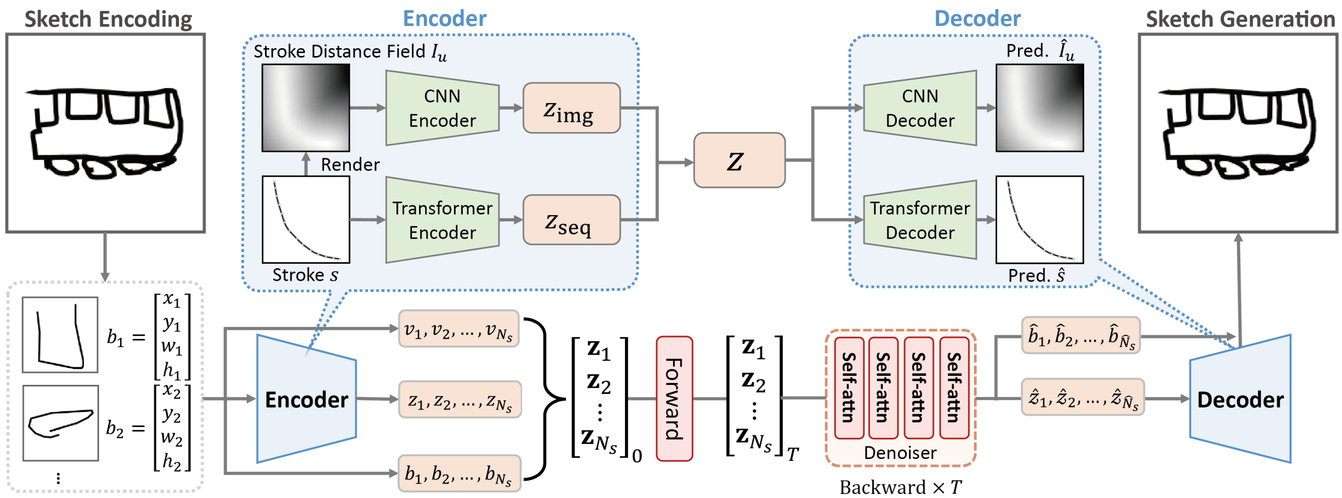

Figure 1: The proposed **StrokeFusion** framework comprises two core components: 1) **Dual-Modal Stroke Encoding**: Each stroke s is processed through parallel encoding paths - a transformer-based sequence encoder handles geometric coordinates while a CNN processes the stroke distance field I_n. These modalities are fused into joint features f, trained via symmetric decoder networks that reconstruct both the original stroke (s) and distance field (I_n); 2) **Sketch Diffusion Generation**: All normalized strokes are encoded into latent vectors z_i, augmented with bounding box parameters $b^i = [x^i, y^i, w^i, h^i]$ and presence flags $v^i \in \{-1, 1\}$. The diffusion model learns the distribution of stroke sequences $\{\mathbf{z}_1, \dots, \mathbf{z}_N\}$ through T-step denoising training. During generation, the denoiser progressively refines noisy latents via reverse diffusion, with valid strokes ($v^i = 1$) being decoded through inverse normalization of $\hat{b}^i$ to reconstruct the final sketch. The architecture maintains permutation invariance through order-agnostic sequence processing.

where γ controls the sharpness of the field. Finally, the unsigned distance field $I_u(g)$ is defined as the maximal contribution over all segments:

$$I_u(g) = \max_{i \in 1, \dots, N_{p-1}} d_i(g). \tag{4}$$

Stroke Embedding

We adopt an encoder-decoder architecture to obtain neural embeddings for individual strokes. The dual-modal encoder jointly processes vector sequences and distance fields through two specialized branches, followed by feature fusion, as illustrated in Figure 1.

Preprocessing

Our two-stage normalization process ensures geometric consistency:

- **Sketch-level normalization**: center and scale the input sketches to $[-1, 1]$ while preserving their aspect ratios. This step standardizes the distribution of stroke bounding boxes.
- **Stroke-level normalization**: independently normalize each stroke's bounding box coordinates to $[0, 1]$ through linear scaling, maintaining individual aspect ratios for geometric relationship preservation.
- **Distance field rendering**: When rendering the unsigned distance field $I_u(g)$ using Equation 4, each stroke is additionally scaled by a factor of 0.8. This margin ensures that strokes do not get truncated by the canvas boundaries during rasterization.

Vector Encoder

Given a stroke point sequence $s_j = p_1, \dots, p_{N_p}$, each point $p_i = (x_i, y_i)$ is first projected via:

$$h_i^0 = \text{Linear}(x_i, y_i) + \text{PE}(i), \tag{5}$$

where $\text{Linear}(\cdot)$ maps coordinates to $\mathbb{R}^{d_h}$ and $\text{PE}(\cdot)$ denotes sinusoidal positional encoding (Vaswani et al. 2017). These embeddings h_i^0 are processed through 6 stacked Transformer layers:

$$h_i^{l+1} = \text{TransformerLayer}(h_i^l). \tag{6}$$

The final output sequence h_i^6 is pooled via attention or average pooling to form a compact sequence-level feature $z_{\text{seq}} \in \mathbb{R}^{d_{\text{seq}}}$.

Image Encoder

The unsigned distance field I_u is encoded through 6 convolutional blocks with progressively increasing channel widths (from 4 to 128), each followed by ReLU activation:

$$z_{\text{img}}^{(l+1)} = \text{ReLU}(\text{Conv2d}(z_{\text{img}}^{(l)})). \tag{7}$$

A global average pooling and linear projection compress the resulting feature map into a visual embedding $z_{\text{img}} \in \mathbb{R}^{d_{\text{img}}}$.

Feature Fusion

The final fused representation z_f combines geometry and visual semantics:

$$z_f = \text{FC}(z_{\text{seq}} \| z_{\text{img}}), \tag{8}$$

where $\|$ denotes concatenation along the feature dimension and $\text{FC}(\cdot)$ is a fully connected layer projecting to $\mathbb{R}^{d_f}$.

Stroke Decoders

Mirroring the encoder, the decoder comprises two parallel branches. The vector decoder expands $\hat{z}_p$ into per-point features using a feedforward network and applies 6 Transformer layers to reconstruct point sequences $\hat{s} = \hat{p}_1, ..., \hat{p}_{N_p}$, where $\hat{p}_i = (\hat{x}_i, \hat{y}_i, \hat{m}_i)$. In parallel, the image decoder upsamples $\hat{z}_u$ through 6 transposed convolutions to produce the reconstructed distance field $\hat{I}_u$. At inference time, only the vector decoder is used to generate strokes with actual drawable trajectories.

Loss Functions

Vector-level Supervision. We apply L_2 loss over coordinate pairs:

$$\mathcal{L}_{\text{vec}} = \frac{1}{N_p} \sum_{i=1}^{N_p} |(x_i, y_i) - (\hat{x}_i, \hat{y}_i)|_2. \tag{9}$$

Image-level Supervision. The distance field output is optimized via:

$$\mathcal{L}_{\text{img}} = |I_u - \hat{I}_u|_2 + \mathcal{L}_{\text{percep}}(I_u, \hat{I}_u), \tag{10}$$

where $\mathcal{L}_{\text{percep}}$ denotes the perceptual loss (Zhang et al. 2018). A KL divergence regularization further aligns latent features:

$$\mathcal{L}_{\text{KL}} = D_{\text{KL}}\left(q(z_f|x), |, \mathcal{N}(0, I)\right). \tag{11}$$

Total Loss. The combined training objective is:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{vec}}\mathcal{L}_{\text{vec}} + \lambda_{\text{img}}\mathcal{L}_{\text{img}} + \lambda_{\text{KL}}\mathcal{L}_{\text{KL}}, \tag{12}$$

where λ_{vec}, λ_{img}, and λ_{KL} are weighting hyperparameters. This multi-level supervision enables the model to learn both stroke geometry and perceptual plausibility, forming the basis for high-fidelity and editable sketch generation.

Diffusion-Based Sketch Generation

Building on the disentangled stroke representations learned in the previous stage, we now describe a generative model that synthesizes entire sketches from sets of latent strokes.

Our encoder maps each stroke into a compact latent space. Built on this, the Sketch Diffusion Generator models stroke collections and spatial relationships by generating unordered and variable-length sequences, as illustrated in Figure 1.

Training Process

During training, each normalized stroke is encoded into a latent embedding z_i using the dual-modal encoder. To model visibility and spatial configuration jointly, each embedding is augmented with its bounding box b_i and a visibility flag $v_i \in \{-1, 1\}$, where $v_i = 1$ indicates a valid stroke and $v_i = -1$ denotes absence. The validity flag is introduced exclusively during the diffusion stage and is not supervised or predicted during encoder pretraining.

The bounding box $b_i = (x, y, w, h)$ includes the center position (x, y) and width-height (w, h) of the stroke, all normalized to $[0, 1]$. During normalization and inverse normalization, scaling is performed isotropically using the larger of w and h to preserve the aspect ratio. The resulting composite latent vector is:

$$\mathbf{z}_i = [z_i,\, b_i,\, v_i]. \tag{13}$$

A sequence of such vectors $\{\mathbf{z}_1, ..., \mathbf{z}_{N_s}\}$ is passed into a diffusion model implemented as a 16-layer Transformer without positional encoding, which ensures permutation invariance. The model denoises these representations through T steps in a Markov chain to learn the joint distribution over strokes and their spatial layout. The value N_s is chosen as a conservative upper bound based on the 99th percentile of sketch stroke counts across the dataset; sketches exceeding this threshold are discarded during training.

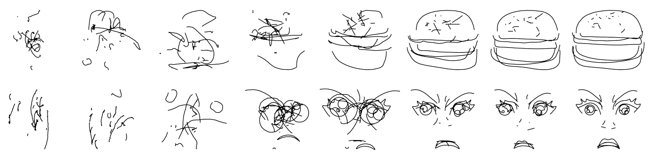

Figure 2: Two examples of the generation process in **StrokeFusion**. From left to right, the noise level progressively decreases. At each timestep, only strokes with presence confidence $\hat{v}_i > 0$ are visualized.

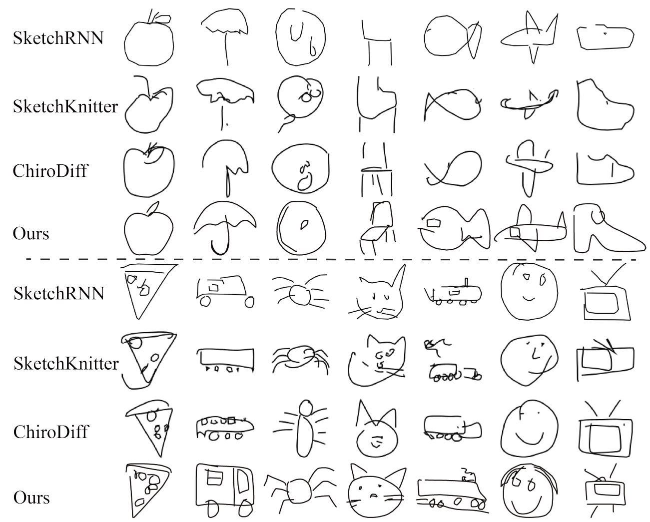

Figure 3: Qualitative comparison of sketches generated by our method and the baselines across different categories in QuickDraw. Our method consistently produces more structurally coherent sketches with richer local details, particularly in complex, multi-stroke scenarios.

Generation Process

At inference time, the diffusion model initializes from a noise sequence $\{\mathbf{z}_1, ..., \mathbf{z}_{N_s}\}_T$ and iteratively denoises it to obtain clean latent representations $\{\mathbf{z}_1, ..., \mathbf{z}_{N_s}\}_0$. Each denoised vector is processed by a shared MLP:

$$[\hat{z}_i,\, \hat{b}_i,\, \hat{v}_i] = \text{MLP}_{\text{split}}(\mathbf{z}_i), \tag{14}$$

where $\hat{v}_i$ is a continuous scalar. During decoding, a stroke is considered present if $\hat{v}_i > 0$, and discarded otherwise.

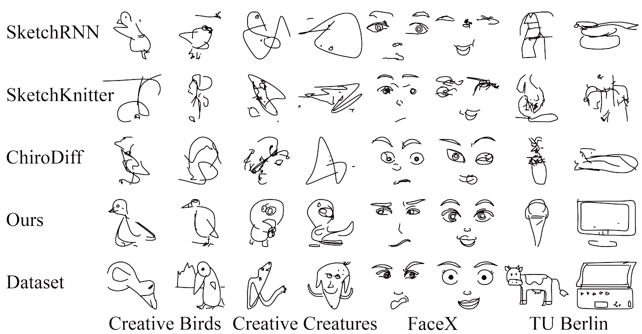

Figure 4: Qualitative generation results on several more complex datasets. Dataset names and representative examples are shown below each column for comparison.

Valid strokes are inversely normalized using $\hat{b}_i$ (with the same isotropic scaling rule) to recover global coordinates. These are then decoded into vector trajectories using the pretrained vector decoder. The resulting sketch is composed by overlaying all retained strokes in their correct positions.

Figure 2 illustrates two examples of the denoising process. As the noise level decreases, each stroke progressively refines its shape, spatial placement, and presence confidence by implicitly referencing other strokes within the composition. This self-organizing behavior emerges from the model's joint reasoning over the entire stroke set.

Unlike existing vector sketch generation methods that treat all strokes as a concatenated trajectory (Yang et al. 2021; Das et al. 2023; Wang et al. 2023), our design allows for a flexible number of strokes and diverse layout patterns. This is achieved by processing strokes as an unordered set without relying on positional encodings, thereby preserving permutation invariance throughout the generation process.

Training Procedure

We follow a two-stage training scheme: (1) pretrain the dual-modal encoder with reconstruction losses, and (2) train the diffusion model to synthesize coherent stroke layouts in latent space. The encoder is frozen in the second stage to ensure consistency.

Experiments

Experimental Setup

Datasets. To evaluate the model's generalization across different sketch and face domains, we use four datasets with varying styles and complexities.

QuickDraw (Ha and Eck 2018) contains over 50 million sketches across 345 categories. Similar to prior work (Wang et al. 2023; Das et al. 2023), we select 14 representative classes (e.g., airplane, face, pizza, spider) for experiments. While widely adopted in prior work, its simplicity makes it insufficient to fully demonstrate our model's strengths, prompting us to include more challenging datasets.

TU Berlin (Eitz, Hays, and Alexa 2012) consists of 20,000 sketches across 250 categories, drawn by non-artists. We use the full set to test the robustness of our method on abstract and stylistically diverse inputs.

FaceX (Cao et al. 2019) provides richly annotated face sketches across attributes such as gender, pose, and expression. We use only front-facing samples to evaluate spatial sensitivity in structured regions like facial features.

Creative (Ge et al. 2020) includes two subsets: Creative Birds and Creative Creatures, with around 10,000 sketches each. Both depict living creatures, with the former limited to birds and the latter covering a wider range of imaginative species. We train on both subsets separately to test adaptability to stylized, creative inputs.

Baseline Methods. We compare our method with three representative vector sketch generation models: SketchRNN (Yang et al. 2021), SketchKnitter (Wang et al. 2023), and ChiroDiff (Das et al. 2023). These three methods all focus on vector sketch generation, avoiding the limitations of traditional raster image modeling by directly outputting stroke sequences. Additionally, we include the set-based raster method, Doodleformer (Bhunia et al. 2022), for comparison.

SketchRNN is an RNN-based variational autoencoder that can generate stroke sequences under conditional or unconditional settings and control generation diversity via a temperature parameter. SketchKnitter introduces a diffusion model framework, treating the generation of stroke points as a progressive denoising process from random initial noise to a clear sketch, significantly improving structural fidelity. ChiroDiff further incorporates a temporal modeling mechanism by resampling strokes, thereby better capturing stroke order and dynamic changes during generation. In contrast, Doodleformer is a raster-based model that predicts bounding boxes for semantic parts (e.g., head/body) and composes them into a final raster image. Its core limitations are that it does not output controllable stroke trajectories and is fundamentally restricted to datasets with semantic part annotations, such as the Creative datasets.

Evaluation Metrics. To evaluate both the quality and diversity of generated sketches, we use three standard metrics: Fréchet Inception Distance (FID) (Heusel et al. 2017), Precision, and Recall (Kynkäänniemi et al. 2019). All metrics are computed over 10,000 sampled images for statistical reliability. To reduce the impact of stroke sampling resolution on feature extraction, we apply the Ramer-Douglas-Peucker (RDP) algorithm (Douglas and Peucker 1973) for stroke simplification before evaluation.

FID measures the distributional distance between real and generated data in the Inception feature space, with lower scores indicating better visual fidelity. Precision reflects the proportion of generated samples that are close to the real data manifold (fidelity), while Recall captures how well the generated samples capture the diversity of the real data. These complementary metrics jointly assess the model's perceptual quality and coverage.

Quantitative Comparison

To systematically evaluate performance across varying sketch complexities, we categorize test classes into three

Method	< 4 strokes			< 8 strokes			≥ 8 strokes		
	FID↓	Prec↑	Rec↑	FID↓	Prec↑	Rec↑	FID↓	Prec↑	Rec↑
SketchRNN	31.61	0.49	0.45	36.98	0.58	0.44	40.67	0.55	0.40
SketchKnitter	23.17	0.52	0.48	27.07	0.57	0.45	35.64	0.54	0.40
ChiroDiff	**17.17**	0.61	0.50	23.84	0.63	0.45	27.78	0.62	0.42
StrokeFusion	19.53	**0.71**	**0.58**	**18.99**	**0.69**	**0.61**	**17.76**	**0.71**	**0.58**

Table 1: Performance comparison across different stroke-count categories in QuickDraw. Classes are grouped by average stroke counts: low-stroke (< 4), medium-stroke (< 8), and high-stroke (≥ 8). Bold and underlined values indicate the best and second-best performances, respectively.

Method	Creative Birds			Creative Creatures			FaceX			TU Berlin		
	FID↓	Prec↑	Rec↑	FID↓	Prec↑	Rec↑	FID↓	Prec↑	Rec↑	FID↓	Prec↑	Rec↑
SketchRNN	59.85	0.26	0.28	121.02	0.44	0.26	155.02	0.01	0.31	98.01	**0.73**	0.20
SketchKnitter	59.16	0.25	0.22	110.46	0.42	0.27	156.97	0.08	0.34	99.46	0.56	0.22
ChiroDiff	60.10	0.56	0.18	36.66	**0.59**	0.27	99.33	0.06	0.30	98.30	0.53	0.25
Doodleformer	27.32	**0.67**	**0.55**	33.46	0.52	**0.69**	-	-	-	-	-	-
StrokeFusion	**26.19**	0.56	0.30	**19.41**	0.58	0.32	**7.27**	**0.76**	**0.89**	**33.68**	0.66	**0.48**

Table 2: Performance comparison on additional datasets. Our method achieves consistently better FID, Precision, and Recall across all datasets. Bold and underlined values indicate the best and second-best performances, respectively.

groups by average stroke count: *low-stroke* ($<$ 4 strokes, including apple, moon, shoe, umbrella, and fish), *medium-stroke* ($<$ 8 strokes, including chair, airplane, television, face, and bus), and *high-stroke* (≥ 8 strokes, including pizza, spider, cat, and train). Table 1 compares our method with the baselines using FID, Precision (Prec), and Recall (Rec), with top-two scores highlighted.

The results show that our method becomes increasingly advantageous as the stroke count grows. For simple sketches, where layout structure is limited, our model, designed to decouple stroke placement and shape generation, has less room to exhibit its strengths, slightly trailing ChiroDiff in some cases. However, as the stroke complexity increases and individual strokes align more with meaningful parts, our structured approach achieves clear improvements in both fidelity and consistency.

To better showcase the structural modeling ability of **StrokeFusion**, we also evaluate on four more challenging datasets: Creative Birds, Creative Creatures, FaceX, and TU Berlin. As shown in Table 2, our method outperforms all baselines in general, confirming strong generalization across styles and domains.

Notably, on FaceX, our gains are especially large. Facial sketches require precise spatial alignment of features like eyes and mouth, which are often distorted by sequential methods due to error accumulation. Our model avoids this by directly predicting stroke positions, leading to more stable and accurate structures.

One exception is SketchRNN's high Precision score on TU Berlin. This likely results from the dataset's large number of categories and limited samples per class, which may cause visually plausible but semantically ambiguous outputs to match real samples in kNN-based evaluation. However, this does not reflect true structural or semantic superiority.

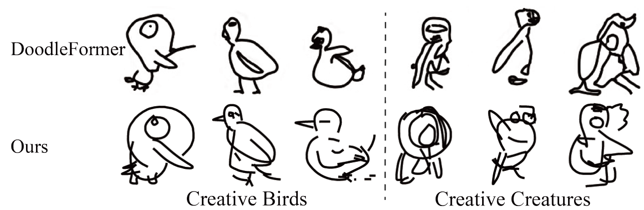

Figure 5: Comparison with DoodleFormer. For each dataset we select visually similar samples from both methods.

Qualitative Comparison

Figure 3 presents the generation results of our method and the baseline methods on the QuickDraw dataset. Our approach captures a broader range of sketch variations and preserves structural diversity, while baseline methods often generate overly simplified or common shapes. In addition, our method better retains local details and demonstrates stronger consistency in global structure.

Figure 4 further shows results on Creative Birds, Creative Creatures, FaceX, and TU Berlin datasets, with ground-truth examples included for reference.

For Creative Birds and Creative Creatures, the high stylistic variation and imaginative designs pose significant challenges. Diffusion-based baselines capture coarse contours but struggle with detail, while SketchRNN suffers from degraded quality in longer sequences. Our method balances structure and detail more effectively, yielding clear and recognizable outputs. We also visually compare our results with Doodleformer (Bhunia et al. 2022), which relies on seman-

Method	moon			television			spider		
	FID↓	Prec↑	Rec↑	FID↓	Prec↑	Rec↑	FID↓	Prec↑	Rec↑
w/o stroke normalization	46.73	0.67	0.04	105.65	0.24	0.03	84.44	0.61	0.02
w/o image-level supervision	38.03	0.57	0.27	46.52	0.53	0.13	139.28	0.11	0.001
The full model	**25.00**	**0.68**	**0.41**	**16.28**	**0.63**	**0.55**	**15.19**	**0.71**	**0.48**

Table 3: Quantitative comparisons of stroke normalization and image-level supervision. Without stroke normalization, the results have degraded sketch fidelity due to inconsistent stroke representations. Without image-level supervision, the sketch quality and the stroke diversity are substantially compromised. Bold values indicate the best performance.

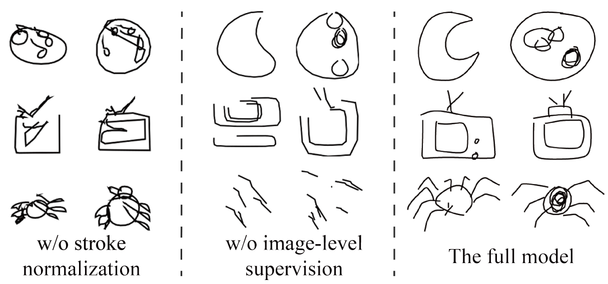

Figure 6: Visualization results of the ablation study. From left to right, the columns correspond to the settings without stroke normalization, without image-level supervision, and our full model. We select three representative categories with varying stroke complexities: Moon (low-stroke), Television (medium-stroke), and Spider (high-stroke).

tic part labels to generate raster sketches (see Figure 5). Our approach achieves comparable visual quality while fundamentally preserving vector representations and stroke-level controllability, unlike the raster output of Doodleformer.

For FaceX, the baseline methods often produce misaligned facial features due to accumulated generation errors. In contrast, our stroke-based modeling yields more regular, semantically aligned results.

For TU Berlin, where class diversity is high and per-class data is limited, the baseline methods tend to produce either fragmented or ambiguous outputs. Our method, while still improvable, consistently produces structurally coherent and semantically interpretable sketches, showing better generalization in low-data settings.

Ablation Study

Stroke Normalization

We normalize strokes by centering and scaling them into a fixed range, ensuring consistent stroke representations and reducing layout variance. To evaluate its effect, we compare with a baseline that directly uses raw strokes without normalization or explicit bounding box input.

As shown in Figure 6 and Table 3, removing normalization significantly degrades performance across both visual quality and quantitative metrics. Particularly, fine structures such as spider legs or television antennas become distorted or misplaced, indicating that normalization is essential for stable stroke encoding and structure consistency.

Image-level Supervision

To evaluate the impact of raster guidance, we ablate the image-level supervision by setting $\gamma = 0$, effectively removing visual input. Without image-level supervision, the model generates strokes with reduced diversity and coherence, especially in high-complexity sketches like *spider*, where outputs degenerate into scattered fragments. This is further reflected in Table 3, where all metrics drop substantially. These results confirm that the visual branch plays a vital role in guiding the encoder toward learning meaningful and diverse stroke patterns.

Limitations

While our method demonstrates strong performance, some limitations persist. First, complex strokes with high-frequency details remain challenging due to their inherent data intensity as point sequences, leading to suboptimal learning efficiency and generation quality for low-stroke sketches. Second, the unsigned distance field (UDF) decoding process introduces computational overhead, as we ultimately rely on point sequences for stroke representation. Future work should explore efficient implicit field decoding schemes or alternative UDF representations to streamline this component.

Additionally, since our method generates sketches with strokes as the minimal unit, it requires that the strokes in the dataset contain semantic information. Vector sketches generated by algorithms often feature strokes that span multiple semantic parts. This structure prevents our model from producing high-quality results. A potential improvement would be to segment input sketches in a more flexible, symbolically semantic manner, allowing better adaptation to such cases.

Conclusion

We have introduced **StrokeFusion**, a novel framework for vector sketch generation that integrates dual-modal stroke encoding with continuous diffusion modeling. Our dual-branch encoder jointly captures geometric and visual features from point sequences and stroke distance fields, fusing them into a unified latent representation. This enriched embedding space facilitates smooth feature transitions during diffusion, enabling the generation of sketches with natural stroke fluidity, structural coherence, and fine-grained detail. Extensive qualitative and quantitative evaluations demonstrate the superiority of **StrokeFusion** in producing diverse, high-quality sketches across varying levels of complexity.

Acknowledgments

We thank the anonymous reviewers for their constructive feedback. This work was partially supported by grants from NSFC (62472287, U21B2023), Guangdong Basic and Applied Basic Research Foundation (2023A1515011297, 2023B1515120026), Shenzhen Natural Science Foundation (JCYJ20250604181519025), ICFCRT (W2441020), and the Scientific Development Funds from Shenzhen University.

References

Alaniz, S.; Mancini, M.; Dutta, A.; Marcos, D.; and Akata, Z. 2022. Abstracting sketches through simple primitives. In *European Conference on Computer Vision*, 396–412. Springer.

Ashcroft, A.; Das, A.; Gryaditskaya, Y.; Qu, Z.; and Song, Y.-Z. 2023. Modelling complex vector drawings with strokeclouds. In *The Twelfth International Conference on Learning Representations*.

Austin, J.; Johnson, D. D.; Ho, J.; Tarlow, D.; and Van Den Berg, R. 2021. Structured denoising diffusion models in discrete state-spaces. *Advances in neural information processing systems*, 34: 17981–17993.

Bandyopadhyay, H.; Bhunia, A. K.; Chowdhury, P. N.; Sain, A.; Xiang, T.; Hospedales, T.; and Song, Y.-Z. 2024. Sketchinr: A first look into sketches as implicit neural representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12565–12574.

Bhunia, A. K.; Khan, S.; Cholakkal, H.; Anwer, R. M.; Khan, F. S.; Laaksonen, J.; and Felsberg, M. 2022. Doodleformer: Creative sketch drawing with transformers. In *European Conference on Computer Vision*, 338–355. Springer.

Cao, N.; Yan, X.; Shi, Y.; and Chen, C. 2019. AI-sketcher: a deep generative model for producing high-quality sketches. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, 2564–2571.

Das, A.; Yang, Y.; Hospedales, T.; Xiang, T.; and Song, Y.-Z. 2020. Béziersketch: A generative model for scalable vector sketches. In *European conference on computer vision*, 632–647. Springer.

Das, A.; Yang, Y.; Hospedales, T. M.; Xiang, T.; and Song, Y.-Z. 2021. Cloud2curve: Generation and vectorization of parametric sketches. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7088–7097.

Das, A.; Yang, Y.; Hospedales, T. M.; Xiang, T.; and Song, Y.-Z. 2022. Sketchode: Learning neural sketch representation in continuous time. In *Tenth International Conference on Learning Representations 2022*.

Das, A.; Yang, Y.; Hospedales, T. M.; Xiang, T.; and Song, Y.-Z. 2023. ChiroDiff: Modelling chirographic data with Diffusion Models. In *International Conference on Learning Representations*.

Dhariwal, P.; and Nichol, A. 2021. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34: 8780–8794.

Douglas, D. H.; and Peucker, T. K. 1973. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: the international journal for geographic information and geovisualization*, 10(2): 112–122.

Eitz, M.; Hays, J.; and Alexa, M. 2012. How do humans sketch objects? *ACM Transactions on graphics (TOG)*, 31(4): 1–10.

Ge, S.; Goswami, V.; Zitnick, C. L.; and Parikh, D. 2020. Creative sketch generation. In *International Conference on Learning Representations*.

Ha, D.; and Eck, D. 2018. A neural representation of sketch drawings. In *International Conference on Learning Representations*.

Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; and Hochreiter, S. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33: 6840–6851.

Kipf, T. N.; and Welling, M. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*.

Kynkäänniemi, T.; Karras, T.; Laine, S.; Lehtinen, J.; and Aila, T. 2019. Improved precision and recall metric for assessing generative models. *Advances in neural information processing systems*, 32.

Lin, H.; Fu, Y.; Xue, X.; and Jiang, Y.-G. 2020. Sketch-bert: Learning sketch bidirectional encoder representation from transformers by self-supervised learning of sketch gestalt. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6758–6767.

Qi, Y.; Su, G.; Wang, Q.; Yang, J.; Pang, K.; and Song, Y.-Z. 2022. Generative sketch healing. *International Journal of Computer Vision*, 130(8): 2006–2021.

Ribeiro, L. S. F.; Bui, T.; Collomosse, J.; and Ponti, M. 2020. Sketchformer: Transformer-based representation for sketched structure. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 14153–14162.

Sohl-Dickstein, J.; Weiss, E.; Maheswaranathan, N.; and Ganguli, S. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, 2256–2265. pmlr.

Tiwari, A.; Biswas, S.; and Lladós, J. 2024. SketchGPT: Autoregressive Modeling for Sketch Generation and Recognition. In *International Conference on Document Analysis and Recognition*, 421–438. Springer.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, J.; Cui, Z.; and Li, C. 2025. VQ-SGen: A Vector Quantized Stroke Representation for Creative Sketch Generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Wang, Q.; Deng, H.; Qi, Y.; Li, D.; and Song, Y.-Z. 2023. Sketchknitter: Vectorized sketch generation with diffusion models. In *International Conference on Learning Representations*.

Xu, P.; Ruan, B.; Zheng, Y.; and Huang, H. 2024. Sketchformer++: A Hierarchical Transformer Architecture for Vector Sketch Representation. In *International Conference on Computational Visual Media*, 24–41. Springer.

Yang, L.; Zhuang, J.; Fu, H.; Wei, X.; Zhou, K.; and Zheng, Y. 2021. Sketchgnn: Semantic sketch segmentation with graph neural networks. *ACM Transactions on Graphics (TOG)*, 40(3): 1–13.

Zang, S.; Tu, S.; and Xu, L. 2023. Self-organizing a latent hierarchy of sketch patterns for controllable sketch synthesis. *IEEE Transactions on Neural Networks and Learning Systems*.

Zhang, B.; Tang, J.; Niessner, M.; and Wonka, P. 2023. 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models. *ACM Transactions On Graphics (TOG)*, 42(4): 1–16.

Zhang, R.; Isola, P.; Efros, A. A.; Shechtman, E.; and Wang, O. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 586–595.