

# Kinetic Typography Diffusion Model

Seonmi Park<sup>1</sup>, Inhwan Bae<sup>1</sup>, Seunghyun Shin<sup>1</sup>, and Hae-Gon Jeon<sup>\*1</sup>

AI Graduate School, GIST, South Korea

{bluesky1000, inhwanbae, seunghyuns98}@gm.gist.ac.kr, haegonj@gist.ac.kr

<https://seonmip.github.io/kinety>

**Abstract.** This paper introduces a method for realistic kinetic typography that generates user-preferred animatable “text content”. We draw on recent advances in guided video diffusion models to achieve visually-pleasing text appearances. To do this, we first construct a kinetic typography dataset, comprising about 600K videos. Our dataset is made from a variety of combinations in 584 templates designed by professional motion graphics designers and involves changing each letter’s position, glyph, and size (*i.e.*, flying, glitches, chromatic aberration, reflecting effects, etc.). Next, we propose a video diffusion model for kinetic typography. For this, there are three requirements: aesthetic appearances, motion effects, and readable letters. This paper identifies the requirements. For this, we present static and dynamic captions used as spatial and temporal guidance of a video diffusion model, respectively. The static caption describes the overall appearance of the video, such as colors, texture and glyph which represent a shape of each letter. The dynamic caption accounts for the movements of letters and backgrounds. We add one more guidance with zero convolution to determine which text content should be visible in the video. We apply the zero convolution to the text content, and impose it on the diffusion model. Lastly, our glyph loss, only minimizing a difference between the predicted word and its ground-truth, is proposed to make the prediction letters readable. Experiments show that our model generates kinetic typography videos with legible and artistic letter motions based on text prompts.

**Keywords:** Kinetic Typography · Video Diffusion · Visual Text Generation

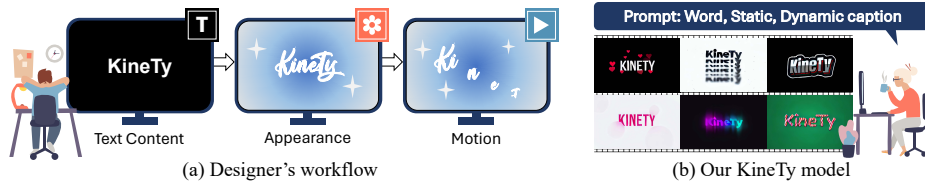
## 1 Introduction

Kinetic typography is an artistic motion graphics design combining text and animations [13]. Based on the word’s meaning, motions are generated to convey information in videos. The main goal is eye-catching and to improve message retention. With the rise of video media, it has become an essential element in TV programs, commercials, music videos and film leaders.

Kinetic typography controls letters’ shape (glyph), color and texture over time, and transforms their positions. Professional motion graphic designers

---

<sup>\*</sup> Corresponding author



**Fig. 1:** An overview of our KineTy pipeline, which is motivated by the designer’s workflow. Our key idea is to generate eye-catching and aesthetic animatable words based on user instructions.

use commercial software like ‘Adobe After Effects’ to make and render kinetic typography videos. The conventional pipeline is as follows [14, 50]; (1) Define a text box in a workspace called composition and enter a text (i.e. word or message). (2) Edit the static text by tuning its font, color and texture (3) Build the background if needed (4) Apply motion effects for all text and background (5) Repeat, optimize, and fine-tune the whole process until it satisfies user intention. This process is time-consuming and labor-intensive. It takes three hours for simple motions to several days for sophisticated effects per single kinetic typography video [14, 26]. To efficiently do this, designers may need to consider either various design references or pre-defined design options called templates.

With the advancement of generative models, there have been attempts to produce dynamic typography. The literature on typography generation has mostly focused on creating static single-letter images. Starting from transforming the shape of the object into a single letter [23, 39, 52, 56], there have been works to create multiple letters and multiple words by introducing a concept of layout [10, 11, 74]. Furthermore, DynTypo [37] and Shape-Matching GAN++ [72] adopt style-transfer techniques to produce animatable effects on a single letter. They focus on only animatable effects without any actual motions of letters.

Meanwhile, the recent advancement of the video generation model has been actively studied with the success of text-guided video diffusion models, especially user’s description conditions video frames [7, 16, 18, 20, 49]. Although these models have opened up the possibility of creating kinetic typography, there is a critical issue that the video generations show a weak understanding of the letters’ shapes and motions.

In this paper, we propose the **Kinetic Typography** diffusion model (called KineTy model) that generates kinetic typography from user-provided text prompts. Inspired by the video diffusion model, we allow users to input comprehensive descriptions for color, font, size, position and motion effects of letters. To better represent text motions, we first introduce our kinetic typography video dataset. 600K videos are rendered by combining randomly generated text contents with 584 templates made by professional motion graphic designers. These videos are labeled with static and dynamic captions that describe the video with respect to its appearance and motion characteristics, respectively.

Next, we present KineTy model that effectively synthesizes videos from the text prompt. Here, we reevaluate how to effectively condition the caption guidance

into the video diffusion model at a fundamental level. We enforce that each effect is incorporated by separately inserting static and dynamic captions into the spatial attention and temporal attention, respectively. To strengthen these attentions, we apply a zero convolution [76] into the word caption, and add it to them. We use a mask loss term only for the generated video by masking out the background contents, which makes the text contents more readable.

Experimental results shows that our KineTy robustly generates kinetic typography video with multi-letter legibility, while accurately representing captions. Furthermore, extensive and meticulous user studies support our claim that KineTy produces more aesthetic outcomes than general-purpose video generations.

## 2 Related Work

### 2.1 Typography Generation

The key of typography is to make readable and visually-appealing text contents to readers. Most research in this field has focused on transferring the style of a single-letter design, including elements like color, glyph, font and effects, to other letters [1, 2, 5, 8, 9, 15, 30–34, 40, 42, 45, 53, 57, 60, 62, 64, 70, 75]. Additionally, there have been works on transferring an image style to the text contents [73]. Similarly, research on scene text editing, which changes text content while maintaining its own style in a scene, is also underway [27, 43, 54, 69], whose extended version to video is available in [51]. In addition to changing text contents, color editing [47] and text segmentation [59, 68] are proposed.

With the recent success of the high-fidelity text-to-image diffusion model [46], typography generation has gained interest. Works in [23, 52, 56] apply image styles into the target letter in an unsupervised manner.

There are concurrent works that display multi-letters, beyond the single-letter generation. They follow the multi-step approach that firstly generates layouts at specific positions, arrange them with multi-letters or multi-words with the same fonts and colors, and contextually infers the background [10, 11, 24, 25, 48, 61, 67, 74, 77]. However, these models are not specialized for typography. Since all letters are generated together, it is not editable for the letters, which makes it difficult to animate and add movement to each character.

### 2.2 Typography Video Generation

As we mentioned above, this work is the first attempt to generate kinetic typography. Although no existing works directly align with this, there are some works related to typography videos [26, 28, 29, 38, 63, 65]. Here, we would like to introduce two methods related to ours.

DynTypo [37] proposes a dynamic typography model that transfers the dynamic effect with realistic movements like fire and water on a specific uppercase English letter to others. Shape-Matching GAN++ [72] transfers an image style into a target letter by matching these shapes with structures of the target letter.

**Table 1:** A summary of the kinetic typography datasets.

Dataset	Domain	#Samples	Video	Static Caption	Dynamic Caption	Multi- letter	Text Appearance	Background	Text Movements
Multi-Content [2]	Typography	10K	-	-	-	-	✓	-	-
TextLogo3K [61]	Typography	3K	-	-	-	-	✓	-	-
TEI41K [71]	Typography	141K	-	-	-	✓	✓	✓	-
TextSeg [68]	Text Segmentation	4K	-	-	-	✓	✓	✓	-
LAION-Glyph-10M [74]	Text-to-Image	10M	-	✓	-	✓	✓	✓	-
MARIO-10M [11]	Text-to-Image	10M	-	✓	-	✓	✓	✓	-
AnyWord-3M [54]	Text-to-Image	3M	-	✓	-	✓	✓	✓	-
CATER-GEN [21]	Image-to-Video	35K	✓	-	✓	N/A	N/A	△	N/A
WebVid-10M [3]	Text-to-Video	10M	✓	✓	△	N/A	N/A	✓	N/A
Vimeo25M [58]	Text-to-Video	25M	✓	✓	△	N/A	N/A	✓	N/A
<b>Ours</b>	Kinetic Typography	600K	✓	✓	✓	✓	✓	✓	✓

However, the position and glyph of the letter are still fixed, with no actual movement between frames. Even under the condition of a single uppercase letter, representing various static and dynamic effects through text prompts remains a challenging issue.

By substituting the motion description with external user input, the task becomes more tractable. A work, Wakey-Wakey [66], transfers the source GIF (Graphics Interchange Format) motion to the target text. Here, graphic designers manually assign additional corresponding key points to achieve motion transfer. However, this process still requires direct human intervention.

### 2.3 Text-to-Video Diffusion Models

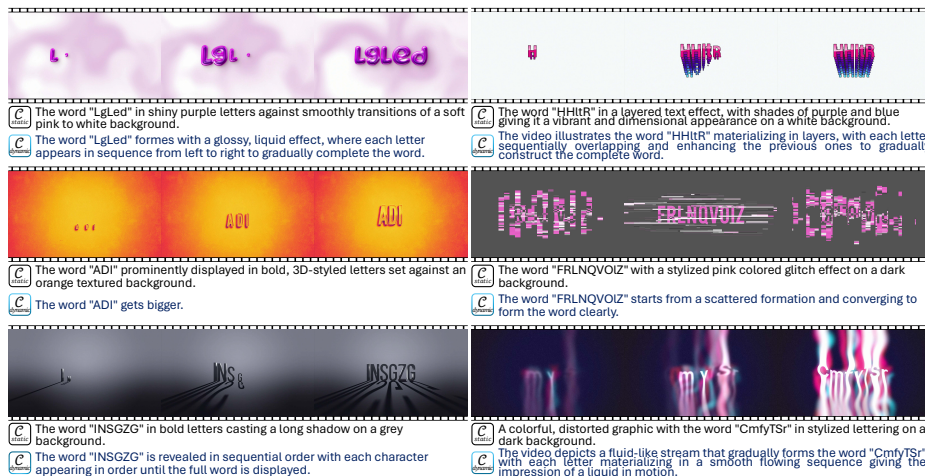
The video diffusion model produces a visually-plausible and photo-realistic video based on text conditioning [7, 12, 16, 18, 20, 36, 49, 58]. The main challenge in the video diffusion model is to maintain the temporal coherency between frames. The pioneer works for video diffusion is to add temporal blocks to the text-to-image model and to learn temporal coherency between video frames [16, 20, 49]. Another type of relevant studies use pretrained models to leverage the temporal block [6, 7, 18]. In particular, Guo *et al.* [18] propose a motion module as the temporal block. It is designed as a plug-and-play module. After joint training with the pre-trained weight of stable diffusion [46], the module enables any text-to-image model to generate the video.

Existing typography has been mainly studied using static single letters. The similar work to kinetic typography is also single letters, with only variation of motion effect and almost no letter’s movement exists. To initiate kinetic typography generation, we present our KineTy dataset and model that allows dynamic motion, effects and glyph deformation of multi-letters, which will be explained at the next section.

## 3 Kinetic Typography Dataset

We describe how to build the KineTy dataset. Unlike previous datasets that only cover single-lettered images [71], our KineTy dataset has not only visual effects on multiple letters, but also their animations. We first provide detailed process





**Fig. 2:** Examples of our KineTy dataset. Our dataset provides high-quality kinetic typography video created by professional motion graphic designers, along with captions that describe the visual appearance and motion effects. To aid visualization, we provide three frames from each video clip.

to render kinetic typography videos using templates made by professional editors in Sec. 3.1. Next, we introduce an way to describe their appearance and motions through text prompts in Sec. 3.2. Lastly, we explain how to make ground-truth kinetic typography video for strictly fair comparisons in Sec. 3.3. We summarize the difference between ours and existing datasets in Tab. 1.

### 3.1 Video Rendering

**Employing templates for video rendering.** The template is a pre-designed project file which contains a visual effect on letters. Many editors prefer to use the templates because it saves their time and labor cost in practice. Following the best practice, we utilize 584 kinetic typography templates from professional graphic designers for our dataset construction, as visualized in Fig. 2.

**Set of multiple letters.** Next, we utilize the kinetic typography templates and randomly replace the text contents for augmentation. We employ multiple letters, in contrast to existing typography datasets with static single-letters [2, 31, 70]. Text contents are randomly generated by arranging up to 12 letters sampled from a set of 52 letters, including both uppercase and lowercase alphabets. Through this, we can expect rich letter-by-letter effects with various arrangements, while also keeping consistent styles across the multiple letters.

**Speculation.** We render videos with  $1,920 \times 1,080$  resolution for 3 sec., and 1,000 random words from 584 templates. Subsequently, all videos are downsampled to  $512 \times 288$  resolution with 8fps for training. It takes a month to render the whole dataset with four i9 13900KF CPUs and an NVIDIA 2080ti GPU.

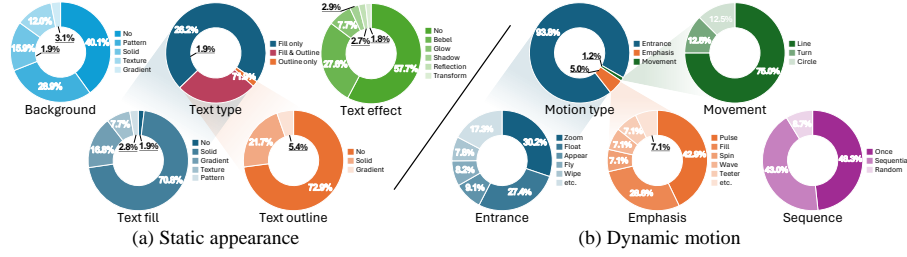


Fig. 3: Statistics of our proposed dataset.

### 3.2 Video Captioning.

**Static and dynamic effect separation.** In the next step, we caption the rendered kinetic typography for text-to-video generation. The professional editors usually design the static appearances at first, and then apply dynamic effects. Following this practice, we use two type of captions: static and dynamic captions, and labeled them accordingly.

**Static effect captioning.** To label static captions, we focus on letter’s appearance which contains spatial information of typography. As shown in Fig. 2, the static captions  $\mathcal{C}_{static}$  describe the color of the letters and the background, the glyph of the letters (e.g., outlined with yellow color or bold font), the characteristics of the background (e.g., textured and shiny background) and the arrangement of the letters (e.g., in a diagonal way) based on the last frame of the video where all the letters are displayed.

**Dynamic effect captioning.** Similarly, we concentrate on the temporal change of motions for each frame to write dynamic captions. Dynamic captions  $\mathcal{C}_{dynamic}$  describe the motion part of the video, such as whether each letter appears in sequential or random order, can be rotated, or has a fade-in effect. For better systematic process, we initially label the videos with the GPT-4Vision model [41], and then manually verify and refine them to fix missing and wrongly labeled components.

**Statistics.** The statistics of our dataset is summarized in Fig. 3. There are three main categories for the static appearance in Fig. 3 (a): Text type, text effect and background. Here, the text fill effects are dominant in the text type because it is the most obvious way to clearly show the text content. Especially, professional designers tend to prefer using solid colors, rather than text outlines. The reason why 57.7% of our dataset has no text effect is the readability of text contents. Since kinetic typography makes dynamic motions of the contents, users do not need for fancy visual effects on it.

In addition, we categorize dynamic motions into two-fold in Fig. 3 (b): motion type, including entrance, emphasis and movement, and sequence type: Our dataset supports various effect of entrance and emphasis to deliver striking messages. In contrast, the line has the majority in the movement because users’ intention is usually conveyed after all letters in a scene are arranged in a line. In the same vein, when words come inside all at one and appear one-by-one in sequence, it is readable. They thus have the high portion in the sequence category.

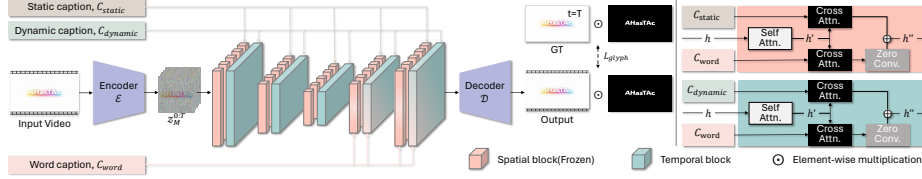


Fig. 4: An architecture of our KineTy model.

### 3.3 Ground-truth Video Generation

Unlike generating random synthetic words in a training phase, we use real-world letters for evaluation by leveraging the templates. We can render ground-truth videos corresponding to user-input captions.

For evaluation, individual letters are selected for each alphabet letter from A to Z, and the first letters are capitalized for checking case sensitivity (*e.g.*, Apple, Ball, ..., Zebra). These 26 words are used for rendering 584 templates, so that a total of 15,184 videos are used for the evaluation.

## 4 Kinetic Typography Diffusion Model

In this section, we present our model for creating kinetic typography videos based on textual prompts. The key concept is to show animateble visual effects on texts, and our challenges in this work are summarized as: (1) the effect should be visually-pleasing and eye-catching to viewers; (2) the transition between frames should be smooth and align with the captions; (3) the text must be readable.

We begin by defining a kinetic typography generation problem in Sec. 4.1. We then describe how to model static and dynamic captions that efficiently guide appearance and motions in Sec. 4.2. We lastly discuss how to improve the glyph legibility with respect to the model design and its learning strategy in Sec. 4.3. Implementation details are provided in Sec. 4.4.

### 4.1 Preliminary

**Conditional Latent Diffusion Models.** Diffusion models train the data distribution by progressively refining a noisy initial state  $z_M \sim \mathcal{N}(0, 1)$  into the target data representation  $z_0$  for  $M$  diffusion steps. Recent advancements, particularly in Latent Diffusion Models (LDMs) [46], enhance the efficiency by encoding an image  $x$  into a compact latent representation  $z_0 = \mathcal{E}(x)$  using an encoder  $\mathcal{E}$ , and is transformed back to the image  $\tilde{x} = \mathcal{D}(z_0)$  using a decoder  $\mathcal{D}$ . These adding noise and its subsequent removal are done with U-Net-style denoising network  $\epsilon_\pi$ . In addition, a condition  $y$  is mapped to the hidden state  $h$  of  $\epsilon_\pi$  via attention:

$$\text{Attn}_\theta(h, y) = \text{Softmax}\left(\frac{Q_\theta(h)K_\theta(y)^\top}{\sqrt{d}}\right) \cdot V_\theta(y) \quad (1)$$

$$\text{s.t. } Q_\theta(h) = W_{\theta,Q} \cdot h, \quad K_\theta(y) = W_{\theta,K} \cdot y, \quad V_\theta(y) = W_{\theta,V} \cdot y,$$

where  $W_{\theta,Q}$ ,  $W_{\theta,K}$  and  $W_{\theta,V}$  denote the learnable parameters for queries, keys and values.  $d$  is the number of dimensions of the keys. The objective function for the denoising network  $\epsilon_\pi$  is formulated as:

$$L_{ldm} = \mathbb{E}_{\mathcal{E}(x), y, \epsilon \sim \mathcal{N}(0,1), m} \left[ \|\epsilon - \epsilon_\pi(z_m, m, y)\|_2^2 \right], \quad (2)$$

where  $\|\cdot\|_2$  is the  $L_2$  distance and diffusion step  $m \in \{1, \dots, M\}$  is uniformly sampled during training.

**Extending image diffusion models to video.** Extending the capability of text-to-image diffusion models to video generation has been more feasible. By incorporating a temporal self-attention called motion module [18], it learns a temporal coherency between  $T$  frames in a latent sequence  $\mathbf{z}_0^{0:T}$ , which corresponds to the video sequence  $\tilde{\mathbf{x}}^{0:T} = \mathcal{D}(\mathbf{z}_0^{0:T})$ . This allows diffusion models to generate smoothly changed sequential images over time, whose temporal self-attention can be defined as follows:

$$\text{Attn}_\phi(\mathbf{h}^{0:T}, \mathbf{h}^{0:T}) = \text{Softmax} \left( \frac{Q_\phi(\mathbf{h}^{0:T}) K_\phi(\mathbf{h}^{0:T})^\top}{\sqrt{d}} \right) \cdot V_\phi(\mathbf{h}^{0:T}). \quad (3)$$

## 4.2 Spatial and Temporal Guidance

**Static caption incorporation.** Motivated by the best practice of professional designers who handle appearance and motion effects separately, we divide the caption into static and dynamic elements.

When existing text-to-image models learn the distribution of image data, they are conditioned on texts related to the image’s appearance. In the same manner, our model is also guided by captions describing the appearance of the text contents and background in each frame of video. Using Eq. (1), we define a spatial attention block with a self-attention followed by a cross-attention with a static caption  $\mathcal{C}_{static}$  as:

$$\mathbf{h}'^{0:T} = \left\{ \text{Attn}_\theta \left( \text{Attn}_\phi(\mathbf{h}^t, \mathbf{h}^t), \tau(\mathcal{C}_{static}) \right) \right\}_{t=0}^{T-1}, \quad (4)$$

where  $\tau(\cdot)$  is the CLIP text encoder [44].

**Dynamic caption incorporation.** A previous work [18] uses a motion module that only utilizes a self-attention between frames to learn the temporal consistency. On the other hand, in kinetic typography, it is essential to accurately display the dynamic motion effects of each letter in a video, following user’s textual description. To do this, we extend Eq. (3) by adding the cross-attention with a dynamic caption as follows:

$$\mathbf{h}'^{0:T} = \text{Attn}_\psi \left( \text{Attn}_\phi(\mathbf{h}^{0:T}, \mathbf{h}^{0:T}), \tau(\mathcal{C}_{dynamic}) \right), \quad (5)$$

Through this process, the model becomes capable of maintaining temporal consistency as well as direct control over dynamic movements.

### 4.3 Enhancing Glyph Legibility.

**Readability improvement.** As a next step, we aim to make the legibility of our model better. To do this, we first classify descriptions for text contents and prompt because diffusion models often have a difficulty in distinguishing them. In this work, we put down a delimiter between each character for text contents.

Let a text content  $\mathcal{L} = \{l_1, l_2, \dots, l_L\}$  contain  $L$  letters. To condition each letter separately, we join each letter  $l_i$  in the text content with a delimiter symbol ‘|’. Further, we add a symbol ‘^’ behind upper case letters because the clip encoder does not care of upper cases. To the end, when the word  $\mathcal{L}$  denotes “Apple”,  $\mathcal{L}'$  is represented as “A^|p|p|l|e”. We denote the whole  $\mathcal{L}$  as  $\mathcal{L}'$  for better representation.

**Attention on text contents.** It can be challenging for the attention module to guide both text contents and effects at once, while disentangling the text content information in CLIP feature space. What is worse, the text content can be overwhelmed by relatively long appearance and motion effects in CLIP text encoder. To tackle this problem, we introduce an additional cross-attention branch to the text content. Inspired by ControlNet [76], we regard the text content as conditions through the zero convolution operation. Starting with the definition of word caption  $\mathcal{C}_{word}$  via a prompt template like “The word  $\{\mathcal{L}'\}$ ”, we extend Eq. (4) by adding the cross-attention module between the word caption and hidden feature, weighted by the zero-initialized convolutions  $\rho$  as follows:

$$\mathbf{h}_{static}''^{0:T} = \left\{ \text{Attn}_\vartheta(\mathbf{h}', \tau(\mathcal{C}_{static})) + \rho_v \left( \text{Attn}_v(\mathbf{h}', \tau(\mathcal{C}_{word})) \right) \right\}_{t=0}^{T-1}, \quad (6)$$

$$\text{where } \mathbf{h}' = \text{Attn}_\theta(\mathbf{h}^t, \mathbf{h}^t).$$

This allows the network to gradually evaluate the usefulness of this additional caption based on the condition. In the same way, we incorporate the word caption into the temporal cross-attention in Eq. (5) as:

$$\mathbf{h}_{dynamic}''^{0:T} = \text{Attn}_\psi(\mathbf{h}', \tau(\mathcal{C}_{dynamic})) + \rho_\varphi \left( \text{Attn}_\varphi(\mathbf{h}', \tau(\mathcal{C}_{word})) \right), \quad (7)$$

$$\text{where } \mathbf{h}' = \text{Attn}_\phi(\mathbf{h}^{0:T}, \mathbf{h}^{0:T}).$$

**Glyph loss.** To train our model, we use a common loss function  $L_{ldm}$ . Additionally, we impose an extra penalty on the letter regions to enforce a sharp and correct glyph of text contents. We first use a binary mask  $B$  for the letters in the last frame  $V^T$  from a text segmentation model [68]. The mask is then blurred to cover its surrounding effects. We then define a glyph loss based on  $L_{LDM}$  with the additional pixel-wise weighting strategy using the blurred mask as follows:

$$L_{glyph} = \mathbb{E}_{\mathcal{E}(x), y, \epsilon \sim \mathcal{N}(0,1), m} \left[ \|B \odot (\epsilon - \epsilon_\pi(z_m^T, m, y))\|_2^2 \right], \quad (8)$$

where  $\odot$  is the element-wise multiplication.

With the glyph loss, we enhance the legibility of text contents, enabling the precise creation of multi-letter formations. Through the linear combination of both loss terms, we can formulate the final loss function as  $L = L_{ldm} + \alpha L_{glyph}$ . Here, we empirically set  $\alpha$  to 0.01.

#### 4.4 Implementation Details

To train our model, we use a two-step training strategy similar to that of human trainees [14]. First, we pre-train the network to generate spatial appearances using only static caption and last video frame pairs. Here, we detach the temporal attention modules to make them work as text-to-image diffusion models during 30 epochs with a batch size of 200. Next, we train the full model while freezing the spatial attention using whole video frames and captions for an epoch with a batch size of 8. For training, we resize the height  $H$  and width  $W$  of the video into  $256 \times 256$  with  $T = 24$  frames. The training is performed with AdamW optimizer [35], with a diffusion step  $M$  of 1000 and a learning rate of 0.0001, which usually takes about 20 hours for training on 8 NVIDIA A100 GPUs. The inference time is about 20 seconds when the number of sampling steps is 25.

### 5 Experiments

In this section, we conduct comprehensive experiments to verify the effectiveness of our model for kinetic typography. We first describe the experimental setup in Sec. 5.1. We then provide comparison results with relevant typography and video generation models, and report user-study to highlight the practical applicability of our model in Sec. 5.2. We lastly carry out an extensive ablation study to validate the effect of each component in our model in Sec. 5.3

#### 5.1 Experimental Setup

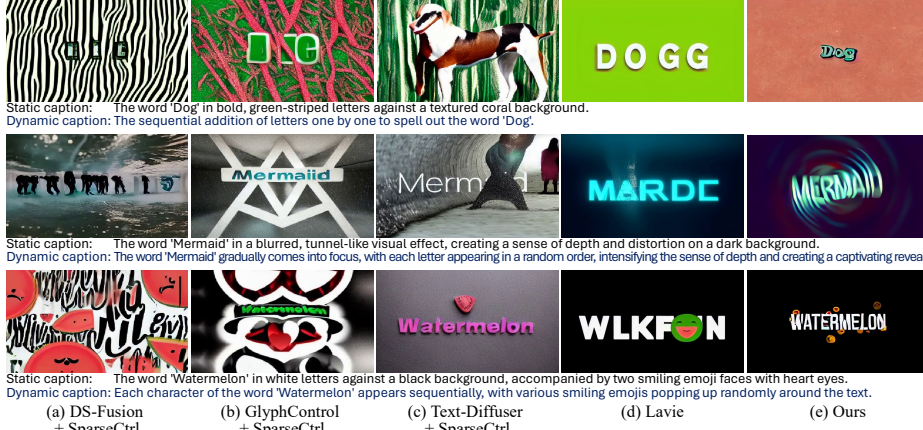
**Comparison methods.** We compare ours with state-of-the-art generative models, which consist of 3 two-stage methods and 2 one-stage methods. The two-stage methods are based on combinations of text-to-image models, including DS-Fusion [52], GlyphControl [74] and Text-Diffuser [11], and an image-to-video model, SparseCtrl [17]. DS-Fusion outputs a stylized letter based on the given style phrase and a letter. Since our caption is a sentence, we extract a style keyword from it and consider it as a style phrase. Note that DS-Fusion is able to generate one letter at a time, so we concatenate each letter to make a full-text content. GlyphControl takes an instruction to generate a glyph image for text contents, then uses the image as a condition for the final output image. Text-Diffuser first finds a proper layout for words and produce the output based on a caption and the layout. Since the comparison methods yield images, we combine them with an image-to-video model, SparseCtrl, to compare with ours. SparseCtrl uses a caption and a guidance such as sketches, depth maps and images to generate a stylized video.

On the other hand, one-stage-methods, AnimateDiff [18] and Lavie [58], are based on text-to-video models. AnimateDiff extends pre-trained text-to-image diffusion models with a motion module, and trains only the motion module to make fully use of the massive information of video dataset. Lavie leverages a temporal self-attention block to enhance a temporal consistency between frames while keeping the generation performance.



**Table 2:** Quantitative evaluation of two-stage generation methods, and one-stage text-to-video generation methods. **Bold:** Best, Underline: Second-best.

Model	Entrance				Emphasis				Motion				Average			
	FVD↓	IS↓	CLIP↑	OCR↑	FVD↓	IS↓	CLIP↑	OCR↑	FVD↓	IS↓	CLIP↑	OCR↑	FVD↓	IS↓	CLIP↑	OCR↑
DS-F[52]+Sparse[17]	1636.3	5.71	0.69	0.58	3184.9	5.97	0.73	0.58	1236.5	5.42	0.76	0.56	1843.10	5.66	0.70	0.58
Glyph[74]+Sparse[17]	2415.6	4.30	0.76	0.68	2667.1	5.06	0.76	<u>0.75</u>	1468.6	4.48	0.84	0.80	2361.98	4.46	0.76	0.69
T-Diff[11]+Sparse[17]	2937.2	5.61	0.78	<b>0.90</b>	3107.5	5.16	0.79	<b>0.89</b>	1208.2	4.06	0.82	<u>0.95</u>	2815.6	5.35	0.78	<b>0.90</b>
AnimateDiff[18]	1613.8	4.71	0.59	0.01	2106.0	4.56	0.66	0.03	2160.6	4.51	0.67	0.07	1727.55	4.68	0.61	0.01
Lavie[58]	<u>825.1</u>	<u>2.25</u>	<u>0.82</u>	0.54	<u>1835.4</u>	<u>3.83</u>	<u>0.83</u>	0.41	<u>505.44</u>	<u>3.33</u>	<u>0.85</u>	0.35	<u>956.97</u>	<u>2.68</u>	<u>0.82</u>	0.49
Ours	<b>147.4</b>	<b>1.79</b>	<b>0.87</b>	<u>0.71</u>	<b>177.52</b>	<b>1.75</b>	<b>0.88</b>	0.54	<b>36.16</b>	<b>1.62</b>	<b>0.94</b>	<b>0.98</b>	<b>125.90</b>	<b>1.77</b>	<b>0.88</b>	<u>0.76</u>

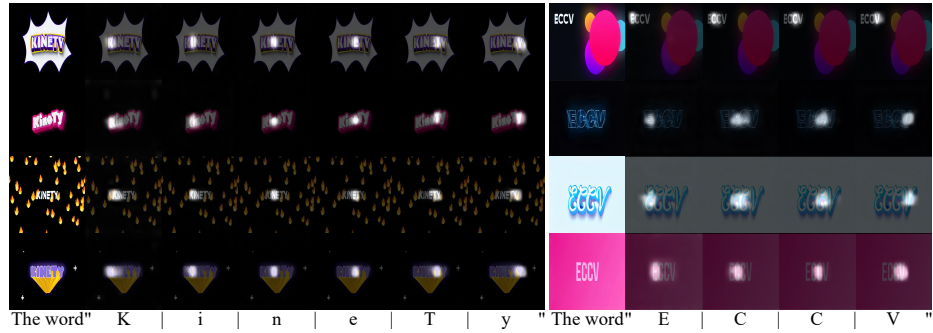
**Fig. 5:** Qualitative results from the comparison models and ours. Obviously, the results from ours reflect the contents of captions better than the others.

For a fair comparison, we utilize the official source codes and pre-trained weights provided by the authors. For more analysis, we categorize the motion effects into three groups, 'Entrance', 'Emphasis' and 'Motion', and conduct experiments to check the quality of each result.

**Evaluation metrics.** Since the final products of kinetic typography depend on the designer's preferences, we adopt four feature-based metrics rather than photo-consistency measures like PSNR. (1) Fréchet Video Distance (FVD) qualifies a similarity between two videos by comparing their feature representations extracted from a pre-trained deep neural network [55]. (2) Inception Score (IS) evaluates the quality and diversity of images generated by a model [4]. (3) CLIPScore measures the semantic alignment between generated images and their textual descriptions, using the CLIP model [19, 22, 44]. (4) Optical Character Recognition (OCR) checks the clarity of the output by comparing recognized text to a reference text using the F1-score.

## 5.2 Evaluation Results

**Quantitative results.** As shown in Tab. 2, our KineTy model achieves the better performance than the comparison methods in the most metrics on all the categories. The comparison methods, despite their impressive performance in general image and video generation, show unsatisfactory performance in creating



**Fig. 6:** Visualization of attention maps between letters from a caption and feature map of a hidden layer.



**Fig. 7:** The demonstration of the editable capability of our KineTy model after generating initial kinetic typography.

kinetic typography. The reasons are mainly two-folds: (1) Text-to-video models are not specialized for kinetic typography and often fail to exactly distinguish the text content in the long captions. Different from them, we additionally feed  $\mathcal{C}_{word}$  which imposes a high attention to the text content in the video. (2) They have challenges in creating letter-level motions. Since they are unaware of each letter in the text content, all the letters move as a whole or inconsistently. We handle this issue by providing a delimiter between each letter and using zero-convolution in Fig. 6.

Our model shows promising results in generating legible text content, but it sometimes falls short in OCR accuracy. This is because the two-stage methods algorithmically generate a glyph image of a text content, and then deform glyphs conditioned on the glyph image. As a result, they have the better performance in OCR metrics than ours and one-stage methods. Nevertheless, our method outperforms in the most cases, demonstrating the effectiveness of our method in generating the clear text content and motion effects.

**Qualitative results.** In Fig. 5, we display three examples of the comparison methods. Compared to ours, the two-stage methods do not understand the user’s instruction with respect to either the whole caption or the letter’s movement. Here, we need to know how to perform our KineTy model. Figure 6 visualizes the attention map between the caption and features from the hidden state on the



**Table 3:** The result of user-study. The score ranges from 1  $\sim$  6 for Study 1 & 2, and 0  $\sim$  1 for Study 3. **Bold:** Best, Underline: Second-best.

User Study	Dynamic Effect	General						Expert					
		DS-F +Sparse	Glyph +Sparse	T-Diff +Sparse	Animate Diff	Lavie	Ours	DS-F +Sparse	Glyph +Sparse	T-Diff +Sparse	Animate Diff	Lavie	Ours
Study 1	Entrance	3.384	3.687	<u>3.891</u>	2.406	3.153	<b>4.478</b>	3.481	4.025	<u>4.131</u>	1.019	3.081	<b>5.263</b>
	Emphasis	3.562	3.713	<u>4.119</u>	2.563	2.725	<b>4.319</b>	3.625	4.175	<u>4.225</u>	1.050	2.625	<b>5.300</b>
	Motion	2.363	<u>4.138</u>	4.113	3.338	2.375	<b>4.675</b>	1.700	3.550	4.000	4.000	2.000	<b>5.750</b>
	Average	3.318	3.713	<u>3.923</u>	2.518	3.036	<b>4.494</b>	3.335	3.995	<u>4.090</u>	1.325	2.905	<b>5.350</b>
Study 2	Entrance	3.514	3.775	<u>3.941</u>	2.413	3.270	<b>4.088</b>	3.650	4.081	<u>4.244</u>	1.019	3.106	<b>4.900</b>
	Emphasis	3.531	3.881	<u>4.075</u>	2.269	3.050	<b>4.194</b>	3.725	<u>4.150</u>	4.050	1.025	2.675	<b>5.375</b>
	Motion	2.838	<u>3.950</u>	3.875	3.250	2.850	<b>4.238</b>	2.150	4.000	4.450	3.450	1.950	<b>5.000</b>
	Average	3.439	3.799	<u>3.948</u>	2.460	3.220	<b>4.135</b>	3.485	4.060	<u>4.255</u>	1.265	2.970	<b>4.995</b>
Study 3	Entrance	0.628	0.620	<u>0.652</u>	0.063	0.169	<b>0.725</b>	0.844	0.856	<u>0.887</u>	0.019	0.138	<b>0.969</b>
	Emphasis	0.581	0.606	<u>0.638</u>	0.088	0.106	<b>0.781</b>	0.750	0.700	<u>0.900</u>	0.000	0.000	<b>1.000</b>
	Motion	0.075	<u>0.713</u>	0.750	0.362	0.288	<b>0.781</b>	0.000	<u>0.900</u>	<u>0.900</u>	0.500	0.250	<b>1.000</b>
	Average	0.553	0.626	<u>0.658</u>	0.096	0.174	<b>0.744</b>	0.730	0.835	<u>0.880</u>	0.065	0.135	<b>0.975</b>

corresponding images. Thanks to the cross-attention between the word caption and the noisy latent, and the deliminators to separate the words, our model successfully embeds features for each letter in the noise.

Lastly, to show the generality of our KineTy model, we conduct an additional experiment. After generating an initial kinetic typography, our KineTy model allows users to modify its style and motions such as the content color, background, font and motion, and even to combine them. As demonstrated in Fig. 7, ours can render the modified outcomes corresponding to the additional captions.

**User Study.** To assess the practical utility of our results in the field of kinetic typography, we conduct a user study using Amazon MTurk. We ask 20 questionnaires to 50 participants. Since their familiarity of typography can vary significantly, we divide them into two groups: experts and non-experts, consisting of 10 and 40 individuals, respectively. The study involves three sections: (1) caption alignment; (2) kinetic typography suitability; (3) Word readability. We make two video clips by randomly choosing two words for each effect. In total, we make 20 clips for this user study.

In the first study, we measure how well ours and the comparison methods generate videos that align with provided captions. The participants see 6 videos from different models with the same corresponding caption in a random order. They are asked to rank the videos from 1 to 6, and the rankings are subsequently converted into scores, ranging from 6 (Best alignment) to 1 (Worst alignment). The following study evaluates how proper the generated videos are for motion graphic applications. Since some participants might be unfamiliar with kinetic typography, all participants watch 4 example videos from online in advance. After that, they check six videos in a random order and rank them. Scores were assigned based on these rankings, from 6 (the most proper) to 1 (the least proper). The last study assesses how readable the outcomes are. After watching 6 videos, the participants are needed to vote for videos with readable text contents. Of course, multiple voting is available.

The performance of each method is measured based on scores for the first two studies, and on the selection ratio for third one. As demonstrated in Tab. 3, our approach shows the promising performance across all studies. The results

**Table 4:** The result of ablation study. **Bold:** Best, Underline: Second-best.

Model	Entrance				Emphasis				Motion				Average			
	FVD↓	IS↓	CLIP↑	OCR↑	FVD↓	IS↓	CLIP↑	OCR↑	FVD↓	IS↓	CLIP↑	OCR↑	FVD↓	IS↓	CLIP↑	OCR↑
<b>Ours</b>	<b>147.4</b>	<u>1.79</u>	<b>0.87</b>	<b>0.71</b>	<b>177.52</b>	<u>1.75</u>	<b>0.88</b>	<u>0.54</u>	<b>36.16</b>	1.62	<u>0.94</u>	<b>0.98</b>	<b>125.90</b>	1.77	<b>0.88</b>	<b>0.76</b>
–Effect separation	789.8	2.75	0.74	0.49	880.1	2.88	0.75	0.49	607.1	2.54	0.82	0.23	729.8	2.73	0.75	0.48
– $C_{word}$	239.1	<b>1.77</b>	<u>0.86</u>	<u>0.70</u>	247.1	<b>1.73</b>	<u>0.87</u>	<u>0.54</u>	62.7	<u>1.60</u>	<b>0.95</b>	<u>0.95</u>	201.8	<b>1.74</b>	<u>0.87</u>	0.74
– $L_{glyph}$	<u>188.9</u>	<b>1.77</b>	<u>0.86</u>	<u>0.70</u>	<u>226.8</u>	<u>1.75</u>	<b>0.88</b>	<b>0.55</b>	<u>50.1</u>	<b>1.59</b>	0.93	<b>0.98</b>	<u>160.8</u>	<u>1.75</u>	<b>0.88</b>	<u>0.75</u>

highlight our model’s practical usability in creating kinetic typography, especially for the domain experts who give our model the highest scores.

### 5.3 Ablation Study

**Without caption separation.** First of all, we train our network without dividing static and dynamic caption. The performances drop significantly for every metrics in Tab. 4. Since we train the text-to-image backbone using only static captions, dynamic captions are considered as noisy input. Similarly, the motion module, used to train dynamic motions, is effective when the simple motion guidance is given, compared to using static and dynamic caption together.

**Text contents incorporation.** We observe that text-to-image diffusion models often fail to catch up on text contents in the prompt when a long description is given. We thus use  $C_{word}$  as an additional input to give a strong attention to each letter. As shown in Tab. 4, without  $C_{word}$ , the performance degradation is observed as expected.

**Without glyph loss.** Lastly, we evaluate the performance without  $L_{glyph}$ . The inferior FVD scores come from the blur effect at the edge of the word along with color bleeding effects.

## 6 Conclusion

In this paper, we propose a generative kinetic typography model, named KineTy. To achieve this, we first build a large-scale kinetic typography dataset by collecting 584 templates from professional designers, and use them to train our diffusion-based network. To effectively handle the text prompt, we split it into static and dynamic captions. They are used to directly guide the spatial and temporal cross-attention for better appearance and motion effects, respectively. Our glyph loss also strengthens the legible and artistic letter generation. We lastly demonstrate that our KineTy produces the visually compelling and semantically coherent kinetic typography videos. The extensive user studies further validate the effectiveness of our model with respect to its practical utility.

**Acknowledgements** This research was supported by ‘Project for Science and Technology Opens the Future of the Region’ program through the INNOPOLIS FOUNDATION funded by Ministry of Science and ICT (Project Number: 2022-DD-UP-0312), the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT)(RS-2024-00338439), GIST-MIT Research Collaboration grant funded by the GIST in 2024, and the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2019-0-01842, Artificial Intelligence Graduate School Program (GIST)).

## References

1. Anderson, D., Shamir, A., Fried, O.: Neural font rendering. arXiv preprint arXiv:2211.14802 (2022) [3](#)
2. Azadi, S., Fisher, M., Kim, V.G., Wang, Z., Shechtman, E., Darrell, T.: Multi-content gan for few-shot font style transfer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2018) [3](#), [4](#), [5](#)
3. Bain, M., Nagrani, A., Varol, G., Zisserman, A.: Frozen in time: A joint video and image encoder for end-to-end retrieval. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2021) [4](#)
4. Barratt, S., Sharma, R.: A note on the inception score. arXiv preprint arXiv:1801.01973 (2018) [11](#)
5. Berio, D., Leymarie, F.F., Asente, P., Echevarria, J.: Strokestyles: Stroke-based segmentation and stylization of fonts. ACM Transactions on Graphics (TOG) (2022) [3](#)
6. Blattmann, A., Dockhorn, T., Kulal, S., Mendelevitch, D., Kilian, M., Lorenz, D., Levi, Y., English, Z., Voleti, V., Letts, A., et al.: Stable video diffusion: Scaling latent video diffusion models to large datasets. arXiv preprint arXiv:2311.15127 (2023) [4](#)
7. Blattmann, A., Rombach, R., Ling, H., Dockhorn, T., Kim, S.W., Fidler, S., Kreis, K.: Align your latents: High-resolution video synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2023) [2](#), [4](#)
8. Campbell, N.D., Kautz, J.: Learning a manifold of fonts. ACM Transactions on Graphics (TOG) (2014) [3](#)
9. Chen, C.H., Liu, Y.T., Zhang, Z., Guo, Y.C., Zhang, S.H.: Joint implicit neural representation for high-fidelity and compact vector fonts. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2023) [3](#)
10. Chen, J., Huang, Y., Lv, T., Cui, L., Chen, Q., Wei, F.: Textdiffuser-2: Unleashing the power of language models for text rendering. arXiv preprint arXiv:2311.16465 (2023) [2](#), [3](#)
11. Chen, J., Huang, Y., Lv, T., Cui, L., Chen, Q., Wei, F.: Textdiffuser: Diffusion models as text painters. Proceedings of the Neural Information Processing Systems (NeurIPS) (2023) [2](#), [3](#), [4](#), [10](#), [11](#)
12. Esser, P., Chiu, J., Atighehchian, P., Granskog, J., Germanidis, A.: Structure and content-guided video synthesis with diffusion models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2023) [4](#)
13. Ford, S., Forlizzi, J., Ishizaki, S.: Kinetic typography: issues in time-based presentation of text. In: CHI'97 extended abstracts on Human factors in computing systems, pp. 269–270. ACM Digital Library (1997) [1](#)
14. Fridsma, L., Gyncild, B.: Adobe After Effects CC Classroom in a Book. Adobe Press (2019) [2](#), [10](#)
15. Fu, B., He, J., Wang, J., Qiao, Y.: Neural transformation fields for arbitrary-styled font generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2023) [3](#)
16. Ge, S., Nah, S., Liu, G., Poon, T., Tao, A., Catanzaro, B., Jacobs, D., Huang, J.B., Liu, M.Y., Balaji, Y.: Preserve your own correlation: A noise prior for video diffusion models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2023) [2](#), [4](#)

17. Guo, Y., Yang, C., Rao, A., Agrawala, M., Lin, D., Dai, B.: Sparsectrl: Adding sparse controls to text-to-video diffusion models. arXiv preprint arXiv:2311.16933 (2023) [10](#), [11](#)
18. Guo, Y., Yang, C., Rao, A., Wang, Y., Qiao, Y., Lin, D., Dai, B.: AnimateDiff: Animate your personalized text-to-image diffusion models without specific tuning. arXiv preprint arXiv:2307.04725 (2023) [2](#), [4](#), [8](#), [10](#), [11](#)
19. Hessel, J., Holtzman, A., Forbes, M., Le Bras, R., Choi, Y.: Clipscore: A reference-free evaluation metric for image captioning. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (2021) [11](#)
20. Ho, J., Chan, W., Saharia, C., Whang, J., Gao, R., Gritsenko, A., Kingma, D.P., Poole, B., Norouzi, M., Fleet, D.J., et al.: Imagen video: High definition video generation with diffusion models. arXiv preprint arXiv:2210.02303 (2022) [2](#), [4](#)
21. Hu, Y., Luo, C., Chen, Z.: Make it move: controllable image-to-video generation with text descriptions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022) [4](#)
22. Huang, Y., Xue, H., Liu, B., Lu, Y.: Unifying multimodal transformer for bi-directional image and text generation. In: Proceedings of the 29th ACM International Conference on Multimedia. pp. 1138–1147 (2021) [11](#)
23. Iluz, S., Vinker, Y., Hertz, A., Berio, D., Cohen-Or, D., Shamir, A.: Word-as-image for semantic typography. ACM Transactions on Graphics (TOG) (2023) [2](#), [3](#)
24. Jahanian, A., Liu, J., Lin, Q., Tretter, D., O'Brien-Strain, E., Lee, S.C., Lyons, N., Allebach, J.: Recommendation system for automatic design of magazine covers. In: Proceedings of the 2013 International Conference on Intelligent User Interfaces (IUI) (2013) [3](#)
25. Jia, P., Li, C., Liu, Z., Shen, Y., Chen, X., Yuan, Y., Zheng, Y., Chen, D., Li, J., Xie, X., et al.: Cole: A hierarchical generation framework for graphic design. arXiv preprint arXiv:2311.16974 (2023) [3](#)
26. Kato, J., Nakano, T., Goto, M.: Textalive: Integrated design environment for kinetic typography. In: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI) (2015) [2](#), [3](#)
27. Krishnan, P., Kovvuri, R., Pang, G., Vassilev, B., Hassner, T.: Textstylebrush: Transfer of text aesthetics from a single example. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) (2023) [3](#)
28. Lee, J.C., Forlizzi, J., Hudson, S.E.: The kinetic typography engine: an extensible system for animating expressive text. In: Proceedings of the 15th annual ACM symposium on User Interface Software and Technology (UIST) (2002) [3](#)
29. Lee, J., Jun, S., Forlizzi, J., Hudson, S.E.: Using kinetic typography to convey emotion in text-based interpersonal communication. In: Proceedings of the 6th conference on Designing Interactive systems (DIS) (2006) [3](#)
30. Li, C., Taniguchi, Y., Lu, M., Konomi, S.: Few-shot font style transfer between different languages. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) (2021) [3](#)
31. Li, X., Wu, L., Wang, C., Meng, L., Meng, X.: Compositional zero-shot artistic font synthesis. In: Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI) (2023) [3](#), [5](#)
32. Liu, X., Meng, G., Chang, J., Hu, R., Xiang, S., Pan, C.: Decoupled representation learning for character glyph synthesis. IEEE Transactions on Multimedia (TMM) (2021) [3](#)
33. Liu, Y.T., Guo, Y.C., Li, Y.X., Wang, C., Zhang, S.H.: Learning implicit glyph shape representation. IEEE Transactions on Visualization and Computer Graphics (TVCG) (2022) [3](#)

34. Liu, Y.T., Zhang, Z., Guo, Y.C., Fisher, M., Wang, Z., Zhang, S.H.: Dualvector: Unsupervised vector font synthesis with dual-part representation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2023) [3](#)
35. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: Proceedings of the International Conference on Learning Representations (ICLR) (2018) [10](#)
36. Luo, Z., Chen, D., Zhang, Y., Huang, Y., Wang, L., Shen, Y., Zhao, D., Zhou, J., Tan, T.: Videofusion: Decomposed diffusion models for high-quality video generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2023) [4](#)
37. Men, Y., Lian, Z., Tang, Y., Xiao, J.: Dyntypo: Example-based dynamic text effects transfer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019) [2](#), [3](#)
38. Minakuchi, M., Tanaka, K.: Automatic kinetic typography composer. In: Proceedings of the ACM SIGCHI International Conference on Advances in Computer Entertainment Technology (ACE) (2005) [3](#)
39. Mu, X., Chen, L., Chen, B., Gu, S., Bao, J., Chen, D., Li, J., Yuan, Y.: Fontstudio: Shape-adaptive diffusion model for coherent and consistent font effect generation. arXiv preprint arXiv:2406.08392 (2024) [2](#)
40. Nagata, Y., Iwana, B.K., Uchida, S.: Contour completion by transformers and its application to vector font data. arXiv preprint arXiv:2304.13988 (2023) [3](#)
41. OpenAI: GPT-4V(ision) system card. [https://cdn.openai.com/papers/GPTV\\_System\\_Card.pdf](https://cdn.openai.com/papers/GPTV_System_Card.pdf) (2023) [6](#)
42. Pan, W., Zhu, A., Zhou, X., Iwana, B.K., Li, S.: Few shot font generation via transferring similarity guided global style and quantization local style. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2023) [3](#)
43. Qu, Y., Tan, Q., Xie, H., Xu, J., Wang, Y., Zhang, Y.: Exploring stroke-level modifications for scene text editing. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI) (2023) [3](#)
44. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: Proceedings of the International Conference on Machine Learning (ICML) (2021) [8](#), [11](#)
45. Reddy, P., Zhang, Z., Wang, Z., Fisher, M., Jin, H., Mitra, N.: A multi-implicit neural representation for fonts. Proceedings of the Neural Information Processing Systems (NeurIPS) (2021) [3](#)
46. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022) [3](#), [4](#), [7](#)
47. Shimoda, W., Haraguchi, D., Uchida, S., Yamaguchi, K.: De-rendering stylized texts. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2021) [3](#)
48. Shimoda, W., Haraguchi, D., Uchida, S., Yamaguchi, K.: Towards diverse and consistent typography generation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) (2024) [3](#)
49. Singer, U., Polyak, A., Hayes, T., Yin, X., An, J., Zhang, S., Hu, Q., Yang, H., Ashual, O., Gafni, O., et al.: Make-a-video: Text-to-video generation without text-video data. In: Proceedings of the International Conference on Learning Representations (ICLR) (2023) [2](#), [4](#)
50. Smith, J., Team, A.C.: Adobe After Effects CS6 Digital Classroom. John Wiley & Sons (2012) [2](#)

51. Subramanian, J., Chordia, V., Bart, E., Fang, S., Guan, K., Bala, R., et al.: Strive: Scene text replacement in videos. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2021) [3](#)
52. Tanveer, M., Wang, Y., Mahdavi-Amiri, A., Zhang, H.: Ds-fusion: Artistic typography via discriminated and stylized diffusion. arXiv preprint arXiv:2303.09604 (2023) [2](#), [3](#), [10](#), [11](#)
53. Thamizharasan, V., Liu, D., Agarwal, S., Fisher, M., Gharbi, M., Wang, O., Jacobson, A., Kalogerakis, E.: Vecfusion: Vector font generation with diffusion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2024) [3](#)
54. Tuo, Y., Xiang, W., He, J.Y., Geng, Y., Xie, X.: Anytext: Multilingual visual text generation and editing. In: Proceedings of the International Conference on Learning Representations (ICLR) (2024) [3](#), [4](#)
55. Unterthiner, T., Van Steenkiste, S., Kurach, K., Marinier, R., Michalski, M., Gelly, S.: Towards accurate generative models of video: A new metric & challenges. arXiv preprint arXiv:1812.01717 (2018) [11](#)
56. Wang, C., Wu, L., Liu, X., Li, X., Meng, L., Meng, X.: Anything to glyph: Artistic font synthesis via text-to-image diffusion model. In: SIGGRAPH Asia 2023 Conference Papers (2023) [2](#), [3](#)
57. Wang, C., Zhou, M., Ge, T., Jiang, Y., Bao, H., Xu, W.: Cf-font: Content fusion for few-shot font generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2023) [3](#)
58. Wang, Y., Chen, X., Ma, X., Zhou, S., Huang, Z., Wang, Y., Yang, C., He, Y., Yu, J., Yang, P., et al.: Lavie: High-quality video generation with cascaded latent diffusion models. arXiv preprint arXiv:2309.15103 (2023) [4](#), [10](#), [11](#)
59. Wang, Y., Ye, Y., Mao, Y., Yu, Y., Song, Y.: Self-supervised scene text segmentation with object-centric layered representations augmented by text regions. In: Proceedings of the 30th ACM International Conference on Multimedia (2022) [3](#)
60. Wang, Y., Lian, Z.: Deepvecfont: Synthesizing high-quality vector fonts via dual-modality learning. ACM Transactions on Graphics (TOG) (2021) [3](#)
61. Wang, Y., Pu, G., Luo, W., Wang, Y., Xiong, P., Kang, H., Lian, Z.: Aesthetic text logo synthesis via content-aware layout inferring. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022) [3](#), [4](#)
62. Wang, Y., Wang, Y., Yu, L., Zhu, Y., Lian, Z.: Deepvecfont-v2: Exploiting transformers to synthesize vector fonts with higher quality. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2023) [3](#)
63. Wong, Y.Y.: Temporal typography: a proposal to enrich written expression. In: Proceedings of the Conference Companion on Human Factors in Computing Systems (CHI) (1996) [3](#)
64. Xia, Z., Xiong, B., Lian, Z.: Vecfontsdif: Learning to reconstruct and synthesize high-quality vector fonts via signed distance functions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2023) [3](#)
65. Xie, L., Shu, X., Su, J.C., Wang, Y., Chen, S., Qu, H.: Creating emordle: Animating word cloud for emotion expression. IEEE Transactions on Visualization and Computer Graphics (TVCG) (2023) [3](#)
66. Xie, L., Zhou, Z., Yu, K., Wang, Y., Qu, H., Chen, S.: Wakey-wakey: Animate text by mimicking characters in a gif. In: Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology (2023) [4](#)
67. Xu, C., Zhou, M., Ge, T., Jiang, Y., Xu, W.: Unsupervised domain adaption with pixel-level discriminator for image-aware layout generation. In: Proceedings of the

- IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2023) [3](#)
68. Xu, X., Zhang, Z., Wang, Z., Price, B., Wang, Z., Shi, H.: Rethinking text segmentation: A novel dataset and a text-specific refinement approach. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021) [3](#), [4](#), [9](#)
  69. Yang, Q., Huang, J., Lin, W.: Swaptext: Image based texts transfer in scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020) [3](#)
  70. Yang, S., Liu, J., Lian, Z., Guo, Z.: Awesome typography: Statistics-based text effects transfer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2017) [3](#), [5](#)
  71. Yang, S., Wang, W., Liu, J.: Te141k: artistic text benchmark for text effect transfer. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) (2020) [4](#)
  72. Yang, S., Wang, Z., Liu, J.: Shape-matching gan++: Scale controllable dynamic artistic text style transfer. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) (2021) [2](#), [3](#)
  73. Yang, S., Wang, Z., Wang, Z., Xu, N., Liu, J., Guo, Z.: Controllable artistic text style transfer via shape-matching gan. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2019) [3](#)
  74. Yang, Y., Gui, D., Yuan, Y., Liang, W., Ding, H., Hu, H., Chen, K.: Glyphcontrol: Glyph conditional control for visual text generation. In: Proceedings of the Neural Information Processing Systems (NeurIPS) (2023) [2](#), [3](#), [4](#), [10](#), [11](#)
  75. Yang, Z., Peng, D., Kong, Y., Zhang, Y., Yao, C., Jin, L.: Fontdiffuser: One-shot font generation via denoising diffusion with multi-scale content aggregation and style contrastive learning. arXiv preprint arXiv:2312.12142 (2023) [3](#)
  76. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2023) [3](#), [9](#)
  77. Zhang, S., Ma, J., Wu, J., Ritchie, D., Agrawala, M.: Editing motion graphics video via motion vectorization and transformation. ACM Transactions on Graphics (TOG) (2023) [3](#)