Dynamic Typography: Bringing Text to Life via Video Diffusion Prior

ZICHEN LIU*, The Hong Kong University of Science and Technology, China
YIHAO MENG*, The Hong Kong University of Science and Technology, China
HAO OUYANG, The Hong Kong University of Science and Technology, China
YUE YU, The Hong Kong University of Science and Technology, China
BOLIN ZHAO, The Hong Kong University of Science and Technology, China
DANIEL COHEN-OR†, Tel-Aviv University, Israel
HUAMIN QU†, The Hong Kong University of Science and Technology, China

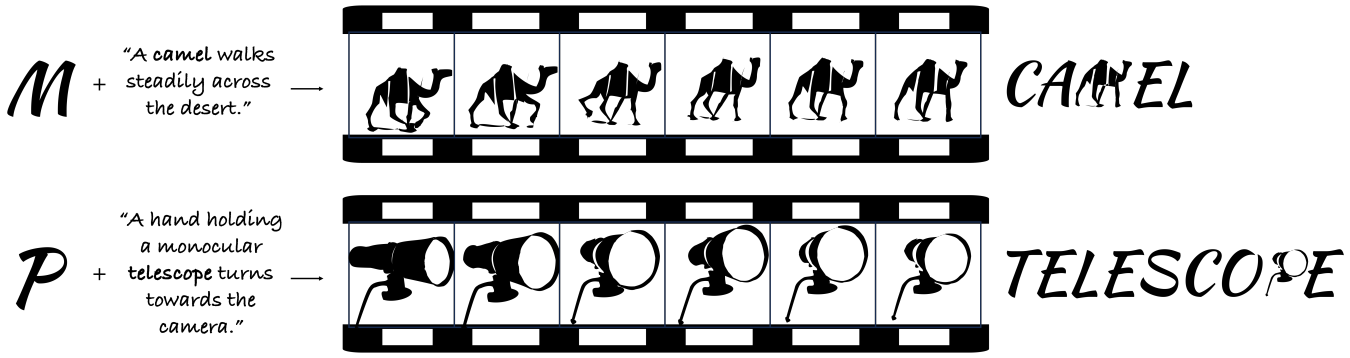


Fig. 1. Given a letter and a text prompt that briefly describes the animation, our method automatically semantically reshapes a letter, and animates it in vector format while maintaining legibility. Our approach allows for a variety of creative interpretations that can dynamically bring words to life.

Text animation serves as an expressive medium, transforming static communication into dynamic experiences by infusing words with motion to evoke emotions, emphasize meanings, and construct compelling narratives. Crafting animations that are semantically aware poses significant challenges, demanding expertise in graphic design and animation. We present an automated text animation scheme, termed "Dynamic Typography", which combines two challenging tasks. It deforms letters to convey semantic meaning and infuses them with vibrant movements based on user prompts. Our technique harnesses vector graphics representations and an end-to-end optimization-based framework. This framework employs neural displacement fields to convert letters into base shapes and applies per-frame motion, encouraging coherence with the intended textual concept. Shape preservation techniques and perceptual loss regularization are employed to maintain legibility and structural integrity throughout the animation process. We demonstrate the generalizability of our approach across various text-to-video models and highlight the superiority of our end-to-end methodology over baseline methods, which might comprise separate tasks. Through quantitative and qualitative evaluations, we demonstrate the effectiveness of our framework in generating coherent text animations that faithfully interpret user prompts while maintaining readability. Our code is available at: https://animate-your-word.github.io/demo/.

1 INTRODUCTION

Text animation is the art of bringing text to life through motion. By animating text to convey emotion, emphasize meaning, and create a dynamic narrative, text animation transforms static messages into vivid, interactive experiences [Lee et al. 2006, 2002b]. The fusion of motion and text, not only captivates viewers, but also deepens the message's impact, making text animation prevail in movies, advertisements, website widgets, and online memes [Xie et al. 2023].

This paper introduces a specialized text animation scheme that focuses on animating individual letters within words. This animation is a compound task: The letters are deformed to embody their semantic meaning and then brought to life with vivid movements based on the user's prompt. We refer to it as "Dynamic Typography". For example, the letter "M" in "CAMEL" can be animated with the prompt "A camel walks steadily across the desert" as illustrated in Fig. 1. This animation scheme opens up a new dimension of textual animation that enriches the user's reading experience.

However, crafting such detailed and prompt-aware animations is challenging, as traditional text animation methods demand considerable expertise in graphic design and animation [Lee et al. 2002b], making them less accessible to non-experts. The technique we present aims to automate the text animation process to make it more accessible and efficient. Following prior research in font generation and stylization [Iluz et al. 2023; Lopes et al. 2019; Wang and

*Denotes equal contribution.
†Denotes corresponding author.

Authors' addresses: Zichen Liu, The Hong Kong University of Science and Technology, Hong Kong, China, zliucz@connect.ust.hk; Yihao Meng, The Hong Kong University of Science and Technology, Hong Kong, China, ymengas@connect.ust.hk; Hao Ouyang, The Hong Kong University of Science and Technology, Hong Kong, China, houyangab@connect.ust.hk; Yue Yu, The Hong Kong University of Science and Technology, Hong Kong, China, yue.yu@connect.ust.hk; Bolin Zhao, The Hong Kong University of Science and Technology, Hong Kong, China, bzhaoan@connect.ust.hk; Daniel Cohen-Or, Tel-Aviv University, Tel Aviv, Israel, cohenor@gmail.com; Huamin Qu, The Hong Kong University of Science and Technology, Hong Kong, China, huamin@ust.hk.

Lian 2021], we represent each input letter and every output frame as a vectorized, closed shape by a collection of Bézier curves. This vector representation is resolution-independent, ensuring that text remains clear and sharp regardless of scale, and brings substantial editability benefits as users can easily modify the text's appearance by adjusting control points. However, this shift to vector graphics introduces unique challenges in text animation. Most current image-to-video methods [Guo et al. 2024; Ni et al. 2023; Wang et al. 2024; Xing et al. 2023] fall short in this new scenario as they are designed to animate rasterized images instead of vectorized shapes, and are hard to render readable text. Although the most recent work, LiveSketch [Gal et al. 2023], introduces an approach to animate arbitrary vectorized sketches, it struggles to preserve legibility and consistency during animation when the input becomes vectorized letters, causing visually unpleasant artifacts including flickering and distortion.

To address these challenges, we designed an optimization-based end-to-end framework that utilizes two neural displacement fields, represented in coordinates-based MLP. The first field deforms the original letter into the base shape, setting the stage for animation. Subsequently, the second neural displacement field learns the per-frame motion applied to the base shape. The two fields are jointly optimized using the score-distillation sampling (SDS) loss [Poole et al. 2023] to integrate motion priors from a pre-trained text-to-video model [Wang et al. 2023], to encourage the animation to align with the intended textual concept. We encode the control point coordinates of the Bézier curve into high-frequency encoding [Mildenhall et al. 2021] and adopt coarse-to-fine frequency annealing [Park et al. 2021] to capture both minute and large motions. To preserve the legibility of the letter throughout the animation, we apply perceptual loss [Zhang et al. 2018] as a form of regularization on the base shape, to maintain a perceptual resemblance to the original letter. Additionally, to preserve the overall structure and appearance during animation, we introduce a novel shape preservation regularization based on the triangulation [Hormann and Greiner 2000] of the base shape, which forces the deformation between the consecutive frames to adhere to the principle of being conformal with respect to the base shape.

Our approach is designed to be data-efficient, eliminating the need for additional data collection or the fine-tuning of large-scale models. Furthermore, our method generalizes well to various text-to-video models, enabling the incorporation of upcoming developments in this area. We quantitatively and qualitatively tested our text animation generation method against various baseline approaches, using a broad spectrum of prompts. The results demonstrate that the generated animation not only accurately and aesthetically interprets the input text prompt descriptions, but also maintains the readability of the original text, outperforming various baseline models in preserving legibility and prompt-video alignment. Overall, our framework demonstrates its efficacy in producing coherent text animations from user prompts, while maintaining the readability of the text, which is achieved by the key design of the learnable base shape and associated shape preservation regularization.

2 RELATED WORK

2.1 Static Text Stylization

Text stylization focuses on amplifying the aesthetic qualities of text while maintaining readability, including artistic text style transfer and semantic typography. Artistic text style transfer aims to migrate stylistic elements from a source image onto text. Existing work incorporates texture synthesis [Fish et al. 2020; Yang et al. 2016] with generative models like GANs [Azadi et al. 2018; Jiang et al. 2019; Mao et al. 2022; Wang et al. 2019]. Semantic typography refers to techniques that blend semantic understanding and visual representation in typography. This encompasses turning letters or words into visual forms that convey their meaning or nature, integrating typography with semantics to enhance the message's clarity and impact. For instance, [Iluz et al. 2023] leverages Score Distillation Sampling [Poole et al. 2023] to deform letters based on the pre-trained diffusion prior [Rombach et al. 2022], encouraging the appearance of the letter to convey the word's semantic meaning. [Tanveer et al. 2023] utilizes a latent diffusion process to construct the latent space of the given semantic-related style and then introduces a discriminator to blend the style into the glyph shape.

These works only produce static images, which in many cases struggle to vividly and effectively communicate meaningful semantic messages. In contrast, our proposed "Dynamic Typograph" infuses text with vibrant motions, which is more effective in capturing the user's attention and giving an aesthetically pleasing impression compared to static text [Minakuchi and Kidawara 2008].

2.2 Dynamic Text Animation

Given the effectiveness of animations in capturing and retaining audience attention [Chang and Ungar 1993], several studies have embarked on designing dynamic text animations. A notable area is dynamic style transfer, which aims to adapt the visual style and motion patterns from a reference video to the target text. Pioneering work by [Men et al. 2019] transferred a style from a source video displaying dynamic text animations onto target static text. [Yang et al. 2021] further enhanced versatility by using a scale-aware Shape-Matching GAN to handle diverse input styles.

Kinetic typography [Ford et al. 1997] represents another innovative direction in text animation, which integrates motion with text to convey or enhance a message. Creating kinetic typography is a labor-intensive and challenging task, motivating many works to reduce the burden and democratize this technique to the public [Ford et al. 1997; Forlizzi et al. 2003; Lee et al. 2002a; Minakuchi and Tanaka 2005]. Recent advancements in AI have enabled a fully automated generation of kinetic typography. For example, [Xie et al. 2023] utilizes a pre-trained motion transfer model [Siarohin et al. 2019] to apply animation patterns from a meme GIF onto text.

However, these approaches require strictly specified driven videos, which are difficult to obtain in real-life scenarios, significantly restricting their usability and generalizability. Moreover, they are constrained to generate specific simple motion patterns, limiting their ability to produce animations with arbitrary complex semantic information. In contrast, our method is generalizable to arbitrary motion patterns and only needs a text prompt as the input.

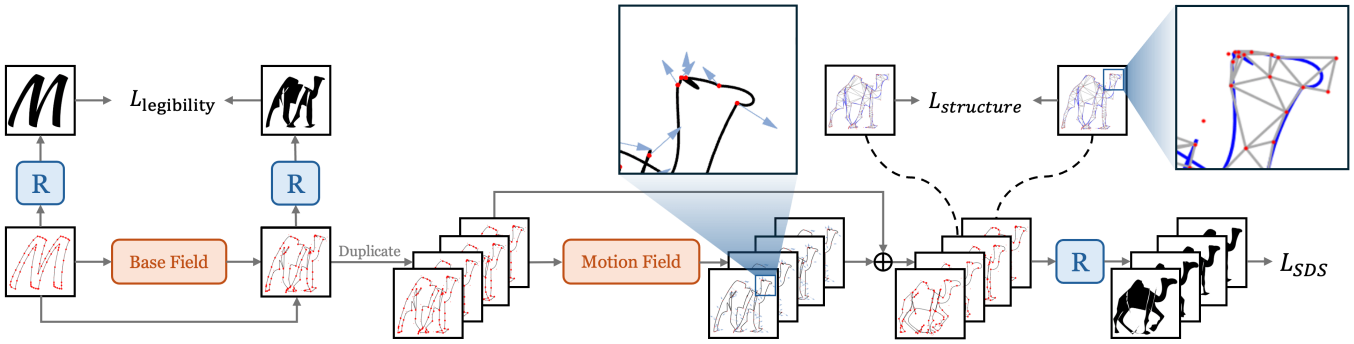


Fig. 2. An overview of the model architecture. Given a letter represented as a set of control points, the Base Field deforms it to the shared base shape, setting the stage to add per-frame displacement. Then we duplicate the base shape across k frames and utilize the Motion Field to predict displacements for each control point at each frame, infusing movement to the base shape. Every frame is then rendered by the differentiable rasterizer R and concatenated as the output video. The base and motion field are jointly optimized by the video prior (L_{SDS}) from frozen pre-trained video foundation model using Score Distillation Sampling, under regularization on legibility $L_{legibility}$ and structure preservation $L_{structure}$.

2.3 Text and Image-to-Video Generation

Text-to-Video generation aims at automatically producing corresponding videos based on textual descriptions. Recent advancements in diffusion models have significantly improved video generation capabilities. Mainstream approaches leverage the power of Stable Diffusion (SD) [Rombach et al. 2022] by incorporating temporal information in a latent space, including AnimateDiff [Guo et al. 2024], LVDM [He et al. 2022], MagicVideo [Zhou et al. 2022], VideoCrafter [Chen et al. 2023] and ModelScope [Wang et al. 2023]. Beyond text-to-video, some methods attempt to generate videos from a given image and a prompt as the condition, such as DynamiCrafter [Xing et al. 2023] and Motion-I2V [Shi et al. 2024]. Several startups also release their image-to-video generation services, e.g., Gen-2 [contributors 2023a], Pika Labs [contributors 2023b], and Stable Video Diffusion (SVD) [Blattmann et al. 2023].

Despite progress, open-source video generation models struggle to maintain text readability during motion, let alone create vivid text animations. Training a model capable of generating high-quality, legible text animations using the aforementioned methods would require a large dataset of text animations, which is difficult to require in practice. One recent work LiveSketch [Gal et al. 2023] introduces an approach to animate arbitrary vectorized sketches without extensive training. This work leverages the motion prior from a large pre-trained text-to-video diffusion model using score distillation sampling [Poole et al. 2023] to guide the motion of input sketches. However, when the input becomes vectorized letters, LiveSketch struggles to preserve legibility and consistency during animation, leading to flickering and distortion artifacts that severely degrade video quality. In contrast, our proposed method successfully generates consistent and prompt-aware text animations while preserving the text readability.

3 PRELIMINARY

3.1 Vector Representation and Fonts

Vector graphics create visual images directly from geometric shapes like points, lines, curves, and polygons. Unlike raster images (like PNG and JPEG), which store data for each pixel, vector graphics are not tied to a specific resolution, making them infinitely scalable and more editable [Ferraiolo et al. 2000].

Hence, modern font formats like TrueType [Penny 1996] and PostScript [Adobe Systems Inc. 1990] utilize vector graphics to define glyph outlines. These outlines are typically collections of Bézier or B-Spline curves, enabling scalable and flexible text rendering, which we aim to preserve. Our method outputs each animation frame in the same vector representations as our input.

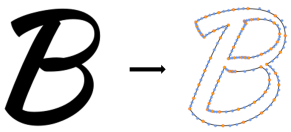


Fig. 3. Bźeier curves representation of letter “B”

In alignment with the setting outlined in [Iluz et al. 2023], we use the FreeType [David Turner 2009] font library to extract the outlines of the specified letter. Subsequently, these outlines are converted into a closed curve composed of several cubic Bézier curves, as illustrated in Fig. 3, to achieve a coherent representation across different fonts and letters.

3.2 Score Distillation Sampling

The objective of Score Distillation Sampling (SDS), originally introduced in the DreamFusion [Poole et al. 2023], is to leverage pre-trained diffusion models' prior knowledge for the text-conditioned generation of different modalities [Katzir et al. 2024]. SDS optimizes the parameters θ of the parametric generator $\mathcal{G}$ (e.g., NeRF [Mildenhall et al. 2021]), ensuring the output of $\mathcal{G}$ aligns well with the prompt. For illustration, assuming $\mathcal{G}$ is a parametric image generator. First, an image $x = \mathcal{G}(\theta)$ is generated. Next, a noise image $z_\tau(x)$ is obtained by adding a Gaussian noise ϵ at the diffusion process's τ-th timestep:

$$z_\tau(x) = \alpha_\tau x + \sigma_\tau \epsilon, \tag{1}$$

where α_τ, and σ_τ are diffusion model's noising schedule, and ϵ is a noise sample from the normal distribution $\mathcal{N}(0, 1)$.

For a pre-trained diffusion model ϵ_ϕ, the gradient of the SDS loss $\mathcal{L}_{SDS}$ is formulated as:

$$\nabla_\phi \mathcal{L}_{SDS} = \left[w(\tau)(\epsilon_\phi(z_\tau(x); y, \tau) - \epsilon) \frac{\partial x}{\partial \theta} \right], \tag{2}$$

where y is the conditioning input to the diffusion model and $w(\tau)$ is a weighting function. The diffusion model predicts the noise added to the image x with $\epsilon_\phi(z_\tau(x); y, \tau)$. The discrepancy between this prediction and the actual noise ϵ measures the difference between the input image and one that aligns with the text prompt. In this work, we adopt this strategy to extract the motion prior from the pre-trained text-to-video diffusion model [Wang et al. 2023].

Since SDS is used with raster images, we utilize DiffVG [Li et al. 2020] as a differentiable rasterizer. This allows us to convert our vector-defined content into pixel space in a differentiable way for applying the SDS loss.

4 METHOD

Problem Formulation. Dynamic Typography focuses on animating individual letters within words based on the user's prompt. The letter is deformed to embody the word's semantic meaning and then brought to life by infusing motion based on the user's prompt.

The original input letter is initialized as a cubic Bézier curves control points set (Fig. 3), denoted as $P_{letter} = \{p_i\}_{i=1}^N = \{(x_i, y_i)\}_{i=1}^N \in \mathbb{R}^{N\times 2}$, where x, y refers to control points' coordinates in SVG canvas, N refers to the total number of control points of the indicated letter. The output video consists of k frames, each represented by a set of control points, denoted as $V = \{P^t\}_{t=1}^k \in \mathbb{R}^{N\cdot k\times 2}$, where P^t is the control points for t-th frame.

Our goal is to learn a displacement for each frame, added on the set of control point coordinates of the original letter's outline. This displacement represents the motion of the control points over time, creating the animation that depicts the user's prompt. We denote the displacement for t-th frame as $\Delta P^t = \{\Delta p_i^t\}_{i=1}^N = \{(\Delta x_i^t, \Delta y_i^t)\}_{i=1}^N \in \mathbb{R}^{N\times 2}$,where Δp_i^t refers to the displacement of the i-th control point in the t-th frame. The final video can be derived as $V = \{P_{letter} + \Delta P^t\}_{t=1}^k$.

To achieve appealing results, we identify three crucial requirements for Dynamic Typography: (1) Temporal Consistency. The deformed letter should move coherently while preserving a relatively consistent appearance in each animation frame. (2) Legibility Preservation. The deformed letter should remain legible in each frame during animation. (3) Semantic Alignment. The letter should be deformed and animated in a way that aligns with the semantic information in the text prompt.

One straightforward strategy can be first deforming the static letter with existing methods like [Iluz et al. 2023], then utilizing an animation model designed for arbitrary graphics composed of Beźier curves like [Gal et al. 2023] to animate the deformed letter. However, this non-end-to-end formulation suffers from conflicting prior knowledge. The deformed letter generated by the first model may not align with the prior knowledge of the animation model. This mismatch can lead the animation model to alter the appearance of the deformed letter, leading to considerable visual artifacts including distortion and inconsistency, see Fig. 4.

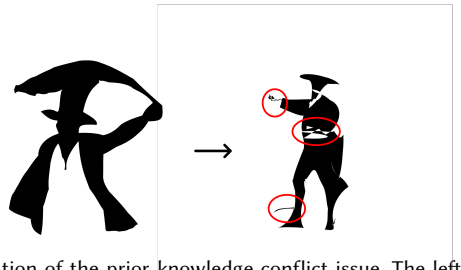


Fig. 4. Illustration of the prior knowledge conflict issue. The left is the deformed "R" for BULLFIGHTER with prompt "A bullfighter holds the corners of a red cape in both hands and waves it" generated by [Iluz et al. 2023], the right is generated by [Gal et al. 2023] to animate the deformed letter with the same prompt. The mismatch in prior knowledge between separate models leads to significant appearance changes and severe artifacts, as highlighted by the red circles.

Therefore, to ensure the coherence of the entire process, we propose an end-to-end architecture that directly maps the original letter to the final animation, as illustrated in Fig. 2. To address the complexity of learning per-frame displacement that converts the input letter into animation, we represent the video as a learnable base shape and per-frame motion added on the base shape (§4.1). Additionally, we incorporate legibility regularization based on perceptual similarity to maintain letter legibility (§4.2). Then, we introduce a mesh-based structure preservation loss to ensure appearance and structure integrity between frames, mitigating issues such as flickering artifacts (§4.3). Finally, we utilize frequency-based encoding and coarse-to-fine annealing to improve the representation of geometry information and motion quality (§4.4).

4.1 Base Field and Motion Field

Learning the per-frame displacement that directly converts the input letter into animation frames is challenging. The video prior derived from foundational text-to-video models using Score Distillation Sampling (SDS) is insufficiently robust to guide the optimization, leading to severe artifacts that degrade the quality of the animation, including distortion, flickering, and abrupt appearance changes in the adjacent frame. Inspired by the CoDeF [Ouyang et al. 2023], we propose modeling the generated video in two neural displacement fields: the base field and the motion field, to address the complexity of this deformation. Both fields are represented by coordinate-based Multilayer Perceptron (MLP). To better capture high-frequency variation and represent geometry information, we project the coordinates into a higher-dimensional space using positional encoding, which is the same as the one used in NeRF [Mildenhall et al. 2021]:

$$\gamma(m) = \left(\sin(2^0\pi m), \cos(2^0\pi m), \ldots, \sin(2^{L-1}\pi m), \cos(2^{L-1}\pi m)\right) \tag{3}$$

$\gamma(\cdot)$ is applied separately to each dimension of the control point coordinates.

The objective of the base field, denoted as B, is to learn a shared shape for every animation frame, serving as a base to infuse motion. It is defined by a function $B : \gamma(P_{letter}) \rightarrow P_B$, which maps the original letter's control points coordinates P_{letter} into base shapes' coordinates P_B, both in $\mathbb{R}^{N\times 2}$.

The motion field, denoted as M, encodes the correspondence between the control points in the base shape and those in each video frame. Inspired by dynamic NeRFs [Park et al. 2021; **?**] and CoDeF [Ouyang et al. 2023], we represent the video as a 3D volume

space, where a control point at t-th frame with coordinate (x, y) is represented by (x, y, t). Specifically, we duplicate the shared base shape k times and encode x, y, and t separately using Eq. 3,writing it as $P'_B : \gamma(\{(P_B, t)\}_{t=1}^{k})$. The motion field is defined as a function $M : P'_B \rightarrow P_V$ that maps control points from the base shape to their corresponding locations P_V in the 3D video space.

To better model motion, we represent P_V as $P_B + \Delta P$, focusing on learning the per-frame displacements $\Delta P = \{\Delta P^t\}_{t=1}^{k}$ to be applied on the base shape. Following [Gal et al. 2023], we decompose the motion into global motion (modeled by an affine transformation matrix shared by all control points of an entire frame) and local motion (predicted for each control point separately). Consider the i-th control point on the base shape with coordinate $(x_{B,i}, y_{B,i})$, its displacement on t-th frame Δp_i^t is summed by its local and global displacement:

$$\Delta p_i^t = \Delta p_i^{t,local} + \Delta p_i^{t,global} \tag{4}$$

$$\Delta p_i^{t,global} = \begin{bmatrix} s_x & sh_x s_y & d_x \\ sh_y s_x & s_y & d_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_{B,i} \\ y_{B,i} \\ 1 \end{bmatrix} - \begin{bmatrix} x_{B,i} \\ y_{B,i} \\ 1 \end{bmatrix}, \tag{5}$$

where $\Delta p_i^{t,local}$ and all elements in the per-frame global transformation matrix are learnable.

To train the base field and motion field, we distill prior knowledge from the large-scale pretrained text-to-video model, using SDS loss of Eq. 2. At each training iteration, we use a differentiable rasterizer [Li et al. 2020], denoted as R, to render our predicted control points set P_V into a rasterized video (pixel format video). We proceed by selecting a diffusion timestep τ, drawing a sample from a normal distribution for noise $\epsilon \sim \mathcal{N}(0, 1)$, and then adding the noise to the rasterized video. The video foundation model denoise this video, based on the user prompt describing a motion pattern closely related to the word's semantic meaning (e.g. "A camel walks steadily across the desert." for "M" in "CAMEL"). The SDS loss is computed in Eq. 2 and guides the learning process to generate videos aligned with the desired text prompt. We jointly optimize the base field and motion field using this SDS loss. The visualized base shape demonstrates alignment with the prompt's semantics, as shown in Fig. 5.

To maintain a legible and consistent appearance throughout the animation, we propose legibility regularization and mesh-based structure preservation regularization, which will be described in later sections.

Fig. 5. Base shape of "Y" for "GYM" with prompt "A man doing exercise by lifting two dumbbells in both hands"

4.2 Legibility Regularization

A critical requirement for Dynamic Typography is ensuring the animations maintain legibility. For example, for "M" in "CAMEL", we hope the "M" takes on the appearance of a camel while being recognizable as the letter "M". When employing SDS loss for training, the text-to-video foundation model's prior knowledge naturally deforms the letter's shape to match the semantic content of the text prompt. However, this significant appearance change compromises the letter's legibility throughout the animation.

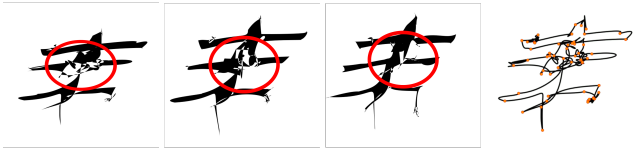
Thus, we propose a regularization term that enforces the letter to be legible, working alongside the SDS loss to guide the optimization process. Specifically, we leverage Learned Perceptual Image Patch Similarity (LPIPS) [Zhang et al. 2018] as a loss to regularize the perceptual distance between the rasterized images of the base shape $R(P_B)$ and the original letter $R(P_{letter})$:

$$\mathcal{L}_{\text{legibility}} = \text{LPIPS}\left(R\left(P_B\right), R\left(P_{letter}\right)\right) \tag{6}$$

Benefiting from our design, we only need to apply this LPIPS-based legibility regularization term to the base shape, and the motion field will automatically propagate this legibility constraint to each frame.

4.3 Mesh-based Structure Preservation Regularization

The optimization process alters the positions of control points, sometimes leading to complex intersections between Bézier curves, as illustrated in Fig. 6d. The rendering of Scalable Vector Graphics (SVG) adheres to the non-zero rule or even-odd rule [Foley 1996], which determines the fill status by drawing an imaginary line from the point to infinity and counting the number of times the line intersects the shape's boundary. The frequent intersections between bezier curves complicate the boundary, leading to alternating black and white "holes" within the image. Furthermore, these intersections between Bézier curves vary abruptly between adjacent frames, leading to severe flickering effects that degrade animation quality, see Fig. 6.



(a) frame 1 (b) frame 2 (c) frame 3 (d) frame 1 vis.

Fig. 6. Adjacent frames of animation for letter "E" in "JET". A large area of alternating black and white "holes" occur within each frame, as highlighted within the red circles, causing severe flickering between the adjacent frames. (d) is the visualization of frame 1, highlighting the control points and the associated Bézier curves. The illustration reveals frequent intersections among the Bézier curves leading to the flickering artifacts.

In addition, the unconstrained degrees of freedom in motion could alter the appearance of the base shape, leading to noticeable discrepancies in appearance between adjacent frames and temporal inconsistency.

To address these issues, we adopt Delaunay Triangulation [Barber and Huhdanpaa 1995; Delaunay et al. 1934] on the base shape based on control points (see Fig. 7). By maintaining the structure of the triangular mesh, we leverage the stability of triangles to prevent frequent intersections between Bézier curves, while also preserving the relative consistency of local geometry information across adjacent frames.

Specifically, we employ the angle variation [Iluz et al. 2023] of the corresponding triangular meshes in adjacent frames as a form of regularization:

$$\frac{1}{k \times m} \sum_{t=1}^{k} \sum_{i=1}^{m} \|\theta(T_{i,t+1}) - \theta(T_{i,t})\|_2, \tag{7}$$

where m refers to the total number of triangular meshes in each frame, $T_{i,t}$ refers to the i-th triangular mesh in the t-th frame, and $\|\theta(T_{i,t+1}) - \theta(T_{i,t})\|_2$ refers to the sum of the squared difference in corresponding angles between two triangles. Particularly, the $(k+1)$-th frame refers to the base shape. During training, we detach the gradients of the second frame in each pair. This allows us to regularize the previous frame's mesh structure with the next frame as a reference, and the last frame is regularized with the base shape as a reference. Hence, the structural constraint with the base shape is propagated to every frame, allowing the preservation of the geometric structure throughout the animation.

We find that this angle-based naive approach is capable of effectively maintaining the triangular structure, thereby alleviating the frequent intersections of the Beźier curves and preserving a relatively stable appearance across different frames without significantly affecting the liveliness of the motion. Furthermore, to ensure that the base shape itself mitigates the frequent intersections of Beźier curves and coarse spikes, we apply the same triangulation-based constraints between the base shape and the input letter. The whole regularization is illustrated in Fig. 7.

$$\begin{aligned} \mathcal{L}_{\text{structure}} =& \frac{1}{m} \sum_{i=1}^{m} \|\theta(T_{i,B}) - \theta(T_{i,\text{letter}})\|_2 \\ &+ \frac{1}{k \times m} \sum_{t=1}^{k} \sum_{i=1}^{m} \|\theta(T_{i,t+1}) - \theta(T_{i,t})\|_2 \end{aligned} \tag{8}$$

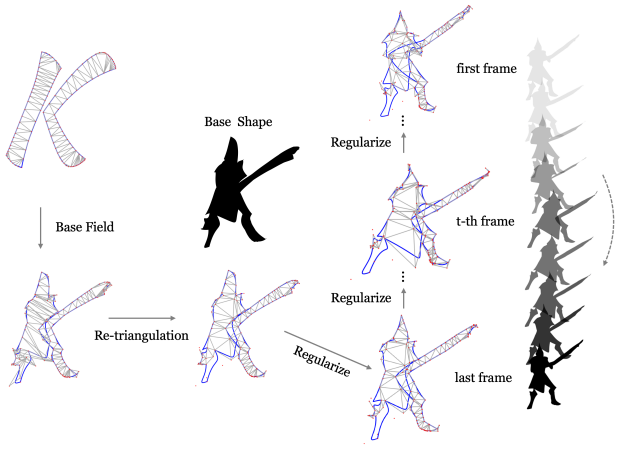


Fig. 7. Illustration of the Mesh-based structure preservation. We first apply this regularization between the base shape and the input letter. We propagate the structural constraint to every frame by regularizing the last frame with the base shape and regularizing every frame with its next frame.

4.4 Frequency-based Encoding and Annealing

NeRF [Mildenhall et al. 2021] have highlighted that a heuristic application of sinusoidal functions to input coordinates, known as "positional encoding", enables the coordinate-based MLPs to capture higher frequency content, as denoted by Eq. 3. We found that this property also applies to our MLPs that use coordinates of control points in Scalable Vector Graphics (SVG) as input. This allows the MLPs in the base and motion field to more effectively represent high-frequency information, corresponding to the detailed geometric features. Additionally, when using coordinate-based MLPs to model motion, a significant challenge is how to capture both minute and large motions. Following Nerfies [Park et al. 2021], we employ a coarse-to-fine strategy that initially targets low-frequency (large-scale) motion and progressively refines the high-frequency (localized) motions. Specifically, we use the following formula to apply weights to each frequency band j in the positional encoding of the MLPs within the motion field.

$$w_j(\alpha) = \frac{1 - \cos(\pi \cdot \text{clamp}(\alpha - j, 0, 1))}{2}, \tag{9}$$

where $\alpha(t) = \frac{Lt}{N}$, t is the current training iteration, and N is a hyper-parameter for when α should reach the maximum number of frequencies L.

In our experiment, this annealed frequency-based encoding resulted in higher-quality motion and detailed geometric information.

5 EXPERIMENTS

To comprehensively evaluate our method's ability, we created a dataset that covers animations for all letters in the alphabet, featuring a variety of elements such as animals, humans, and objects. This dataset contains a total of 33 Dynamic Typography samples. Each sample includes a word, a specific letter within the word to be animated, and a concise text prompt describing the desired animation. We used KaushanScript-Regular as the default font.

We use text-to-video-ms-1.7b model in ModelScope [Luo et al. 2023; Wang et al. 2023] for the diffusion backbone. We apply augmentations including random crop and random perspective to all frames of the rendered videos. Each optimization takes 1000 epochs, about 40 minutes on a single H800 GPU.

To illustrate our method's capabilities, we present some generated results in Fig. 1. These animations vividly bring the specified letter to life while adhering to the prompt and maintaining the word's readability. For further exploration, we strongly suggest the readers go through the additional examples and full-length videos on our project page.

5.1 Comparisons

We compare our method with approaches from two distinct categories: the pixel-based strategies leveraging either text-to-video or image-to-video methods, and the vector-based animation method.

Within the pixel-based scenario, we compare our model against the leading text-to-video generation models Gen-2 [contributors 2023a] (ranked first in the EvalCrafter [Liu et al. 2023] benchmark) – a commercial web-based tool, and DynamiCrafter [Xing et al. 2023], the state-of-the-art model for image-to-video generation conditioned on text. For text-to-video generation, we append the prompt

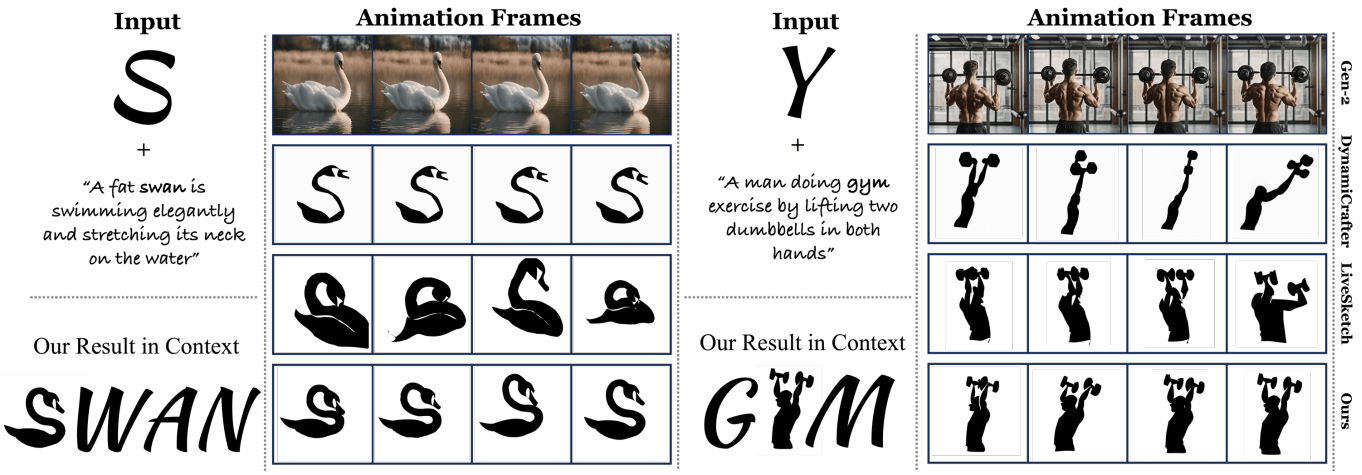


Fig. 8. Visual comparisons between the baselines and our model. Text-to-image model (Gen-2) generates colorful images but fails to maintain the shape of the original letter. The pixel-based image-to-video model (DynamiCrafter) produces results with little, sometimes unreasonable motion. The general vector animation model (LiveSketch) struggles to preserve legibility and maintain a stable appearance across frames.

with "which looks like a letter β," where β represents the specific letter to be animated. In the image-to-video case, we use the stylized letter generated by the word-as-image [Iluz et al. 2023] as the conditioning image.

Within the vector-based scenario, we utilize LiveSketch [Gal et al. 2023] as a framework to animate vector images. To ensure a fair comparison, we condition the animation on the stylized letter generated by the word-as-image [Iluz et al. 2023] as well.

Qualitative Comparison We present the visual comparison with baseline methods in Fig. 8. While achieving high resolution and realism, Gen-2 struggles to generate frames that keep the letter's shape, which greatly harms the legibility. With DynamiCrafter, the "SWAN" animation exhibits minimal movement, while the "GYM" animation features unrealistic motion that deviates from the user's prompt. Although LiveSketch can depict the user's prompt through animation, it sacrifices legibility. Also, the letter's appearance deteriorates throughout the animation, as demonstrated in the "SWAN" example. Our model strikes a balance between prompt-video alignment and letter legibility. It consistently generates animations that adhere to the user's prompt while preserving the original letter's form. This allows the animation to seamlessly integrate within the original word, as showcased by the in-context results in Fig. 8.

Method	Perceptual Input Conformity (↑)	Text-to-Video Alignment(↑)
Gen-2	0.1495	23.3687
DynamiCrafter	0.5151	17.8124
LiveSketch	0.4841	20.2402
Ours	0.5301	21.4391

Table 1. Quantitative results between the baselines and our model. The best and second-best scores for each metric are highlighted in red and blue respectively.

Quantitative Comparison Tab. 1 presents the quantitative evaluation results. We employed two metrics, Perceptual Input Conformity (PIC) and Text-to-Video Alignment. Following DynamiCrafter [Xing et al. 2023], we computed Perceptual Input Conformity (PIC) using DreamSim's [Poole et al. 2023] perceptual distance metric between each output frame and the input letter, averaged across all frames. This metric assesses how well the animation preserves the original letter's appearance. To evaluate the alignment between the generated videos and their corresponding prompts ("text-to-video alignment"), we leverage the X-CLIP score [Ma et al. 2022], which extends CLIP [Radford et al. 2021] to video recognition, to obtain frame-wise image embeddings and text embeddings. The average cosine similarity between these embeddings reflects how well the generated videos align with the corresponding prompts.

While Gen-2 achieves the highest text-to-video alignment score, it severely suffers in legibility preservation (lowest PIC score). Conversely, our model excels in PIC (highest score), indicating the effectiveness in maintaining the original letter's form. While achieving the second-best text-to-video alignment score, our method strikes a balance between faithfully representing both the animation concept and the letter itself.

5.2 Ablation Study

We conducted an ablation study to analyze the contribution of each component in our proposed method: learnable base shape, legibility regularization, mesh-based structure preservation regularization, and frequency encoding with annealing. Visual results in Fig. 9 showcase the qualitative impact of removing each component. Quantitative results in Tab. 2 further confirm their effectiveness.

In addition to Perceptual Input Conformity (PIC) and Text-to-Video Alignment (X-CLIP score), we employed warping error to assess temporal consistency, following EvalCrafter [Liu et al. 2023]. This metric estimates the optical flow between consecutive frames using the pre-trained RAFT model [Teed and Deng 2020] and calculates the pixel-wise difference between the warped image and the

target image. The lower warping error indicates smoother and more temporally consistent animations.

Base Shape The calculation of legibility and structure preservation regularization involves the base shape. Hence, when removing the learnable base shape, the legibility loss $\mathcal{L}_{legibility}$ is computed between every output frame and the input letter, while the structure preservation loss $\mathcal{L}_{structure}$ is applied between every pair of consecutive frames.

As observed in Fig. 9 (row 2), removing the shared learnable base shape results in inconsistent animations. Specifically, as highlighted by the red circle, the appearance of the bullfighter deviates significantly between frames, harming legibility. The finding is also supported by Tab. 2 (row 2), where removing the base shape results in significant degradation under all three metrics.

Legibility Regularization Without the perceptual regularization on the base shape, the base shape struggles to preserve legibility. As a result, each animation frame loses the letter "R" shape in Fig. 9 (row 3), leading to lower PIC in Tab. 2 (row 3).

Structure Preservation Regularization Removing mesh-based structure preservation allows the base shape's structure to deviate from the original letter, causing the discontinuity between the bullfighter and cape in the base shape and all frames, as highlighted in Tab. 2 (row 4). Without this regularization term, the animation shows inconsistent appearances across different frames, which degrades the legibility, leading to the lowest PIC in Tab. 2 (row 4).

Frequency Encoding and Annealing When removing frequency encoding and coarse-to-fine annealing, the motion and geometry quality suffers. For example, the bullfighter animation in Fig. 9 (row 5) shows unreasonable motion and geometry details, resulting in an animation that does not accurately represent the text prompt. Moreover, the degradation in motion quality also harms the temporal consistency, Tab. 2 (row 5).

Method	Optical Flow Warping Error(↓)	Perceptual Input Conformity(↑)	Text-to-Video Alignment(↑)
Full Model	**0.01645**	**0.5310**	**21.4447**
No Base Shape	0.03616	0.5178	20.0568
No Legibility	**0.01561**	0.4924	20.2857
No Struc. Pre.	0.01777	0.4906	20.6285
No Freq.	0.02222	**0.5377**	**20.8280**

Table 2. Quantitative results of the ablation study. The best and second-best scores for each metric are highlighted in red and blue respectively.

5.3 Generalizability

Our optimization framework, leveraging Score Distillation Sampling (SDS), achieves generalization across various diffusion-based text-to-video models. To demonstrate this, we applied different base models for computing $\mathcal{L}_{SDS}$, including the 1.7-billion parameter text-to-video model from ModelScope [Wang et al. 2023], AnimateDiff [Guo et al. 2024], and ZeroScope [Luo et al. 2023]. Fig. 10 presents visual results for the same animation sample ("Knight") with each base model.

While the letter "K" exhibits deformations and animation styles unique to each model, all animations accurately depict the user's

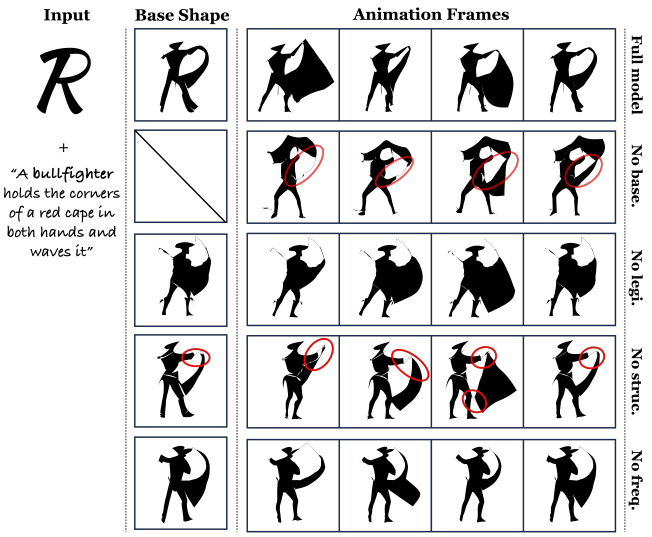


Fig. 9. Visual comparisons of ablation study. Removing base shape or structure preservation regularization results in shape deviation and flickering. Without legibility regularization, each animation frame loses the letter "R" shape. The absence of frequency encoding and annealing leads to the degradation of the motion quality and geometry details.

prompt and maintain the basic "K" shape. This showcases the generalizability of our method. Hence, future advancements in text-to-video models with stronger prior knowledge will benefit our approach.

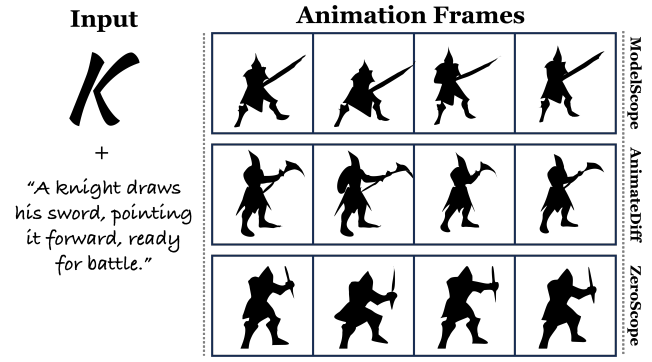


Fig. 10. Visual results of the same animation sample using different text-to-video base models.

6 CONCLUSION

We propose an automated text animation scheme, termed "Dynamic Typography," that deforms letters to convey semantic meaning and animates them vividly based on user prompts. Our method is an end-to-end optimization-based approach and is generalizable to arbitrary words and motion patterns. Nevertheless, there remain several limitations. First, the motion quality can be bounded by the video foundation model, which may be unaware of specific motions in some cases. However, our framework is model-agnostic, which facilitates integration with future advancements in diffusion-based video

foundation models. Besides, challenges arise when user-provided text prompts deviate significantly from original letter shapes, complicating the model's ability to strike a balance between generating semantic-aware vivid motion and preserving the legibility of the original letter. We hope that our work can open the possibility for further research of semantic-aware text animation that incorporates the rapid development of video generation models.

REFERENCES

Adobe Systems Inc. 1990. *Adobe Type 1 Font Format.* Addison Wesley Publishing Company.

Samaneh Azadi, Matthew Fisher, Vladimir Kim, Zhaowen Wang, Eli Shechtman, and Trevor Darrell. 2018. Multi-Content GAN for Few-Shot Font Style Transfer. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition.* https://doi.org/10.1109/cvpr.2018.00789

C Barber and Hannu Huhdanpaa. 1995. Qhull. The Geometry Center, University of Minnesota.

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. 2023. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127* (2023).

Bay-Wei Chang and David Ungar. 1993. Animation: from cartoons to the user interface. In *Proceedings of the 6th Annual ACM Symposium on User Interface Software and Technology* (Atlanta, Georgia, USA) *(UIST '93)*. Association for Computing Machinery, New York, NY, USA, 45–55. https://doi.org/10.1145/168642.168647

Haoxin Chen, Menghan Xia, Yingqing He, Yong Zhang, Xiaodong Cun, Shaoshu Yang, Jinbo Xing, Yaofang Liu, Qifeng Chen, Xintao Wang, et al. 2023. Videocrafter1: Open diffusion models for high-quality video generation. *arXiv preprint arXiv:2310.19512* (2023).

Gen-2 contributors. 2023a. Gen-2. https://research.runwayml.com/gen2

PikaLabs contributors. 2023b. Pikalabs. https://www.pika.art/

Werner Lemberg David Turner. 2009. *FreeType library.* Retrieved Mar 19, 2024 from https://freetype.org/

Boris Delaunay et al. 1934. Sur la sphere vide. *Izv. Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk* 7, 793-800 (1934), 1–2.

Jon Ferraiolo, Fujisawa Jun, and Dean Jackson. 2000. *Scalable vector graphics (SVG) 1.0 specification.* iuniverse Bloomington.

Noa Fish, Lilach Perry, Amit Bermano, and Daniel Cohen-Or. 2020. SketchPatch. *ACM Transactions on Graphics* (Dec 2020), 1–14. https://doi.org/10.1145/3414685.3417816

James D Foley. 1996. *Computer graphics: principles and practice.* Vol. 12110. Addison-Wesley Professional.

Shannon Ford, Jodi Forlizzi, and Suguru Ishizaki. 1997. Kinetic typography. In *CHI '97 extended abstracts on Human factors in computing systems looking to the future - CHI '97.* https://doi.org/10.1145/1120212.1120387

Jodi Forlizzi, Johnny Lee, and Scott Hudson. 2003. The kinedit system. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* https://doi.org/10.1145/642611.642677

Rinon Gal, Yael Vinker, Yuval Alaluf, Amit H. Bermano, Daniel Cohen-Or, Ariel Shamir, and Gal Chechik. 2023. Breathing Life Into Sketches Using Text-to-Video Priors. (2023). arXiv:2311.13608 [cs.CV]

Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. 2024. AnimateDiff: Animate Your Personalized Text-to-Image Diffusion Models without Specific Tuning. In *The Twelfth International Conference on Learning Representations.* https://openreview.net/forum?id=Fx2SbBgcte

Yingqing He, Tianyu Yang, Yong Zhang, Ying Shan, and Qifeng Chen. 2022. Latent Video Diffusion Models for High-Fidelity Video Generation with Arbitrary Lengths. (Nov 2022).

Kai Hormann and Günther Greiner. 2000. MIPS: An efficient global parametrization method. *Curve and Surface Design: Saint-Malo 1999* (2000), 153–162.

Shir Iluz, Yael Vinker, Amir Hertz, Daniel Berio, Daniel Cohen-Or, and Ariel Shamir. 2023. Word-As-Image for Semantic Typography. *ACM Trans. Graph.* 42, 4, Article 151 (jul 2023), 11 pages. https://doi.org/10.1145/3592123

Yue Jiang, Zhouhui Lian, Yingmin Tang, and Jianguo Xiao. 2019. SCFont: Structure-Guided Chinese Font Generation via Deep Stacked Networks. *Proceedings of the AAAI Conference on Artificial Intelligence* (Sep 2019), 4015–4022. https://doi.org/10.1609/aaai.v33i01.33014015

Oren Katzir, Or Patashnik, Daniel Cohen-Or, and Dani Lischinski. 2024. Noise-free Score Distillation. In *The Twelfth International Conference on Learning Representations.* https://openreview.net/forum?id=dlIMcmlAdk

Joonhwan Lee, Soojin Jun, Jodi Forlizzi, and Scott E. Hudson. 2006. Using kinetic typography to convey emotion in text-based interpersonal communication. In *Proceedings of the 6th Conference on Designing Interactive Systems* (University Park, PA, USA) *(DIS '06)*. Association for Computing Machinery, New York, NY, USA, 41–49. https://doi.org/10.1145/1142405.1142414

Johnny C. Lee, Jodi Forlizzi, and Scott E. Hudson. 2002a. The kinetic typography engine. In *Proceedings of the 15th annual ACM symposium on User interface software and technology.* https://doi.org/10.1145/571985.571997

Johnny C. Lee, Jodi Forlizzi, and Scott E. Hudson. 2002b. 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* (Paris, France) *(UIST '02)*. Association for Computing Machinery, New York, NY, USA, 81–90. https://doi.org/10.1145/571985.571997

Tzu-Mao Li, Michal Lukáč, Michaël Gharbi, and Jonathan Ragan-Kelley. 2020. Differentiable vector graphics rasterization for editing and learning. *ACM Transactions on Graphics* (Dec 2020), 1–15. https://doi.org/10.1145/3414685.3417871

Yaofang Liu, Xiaodong Cun, Xuebo Liu, Xintao Wang, Yong Zhang, Haoxin Chen, Yang Liu, Tieyong Zeng, Raymond Chan, and Ying Shan. 2023. Evalcrafter: Benchmarking and evaluating large video generation models. *arXiv preprint arXiv:2310.11440* (2023).

Raphael Gontijo Lopes, David Ha, Douglas Eck, and Jonathon Shlens. 2019. A learned representation for scalable vector graphics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision.* 7930–7939.

Zhengxiong Luo, Dayou Chen, Yingya Zhang, Yan Huang, Liang Wang, Yujun Shen, Deli Zhao, Jingren Zhou, and Tieniu Tan. 2023. VideoFusion: Decomposed Diffusion Models for High-Quality Video Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).*

Yiwei Ma, Guohai Xu, Xiaoshuai Sun, Ming Yan, Ji Zhang, and Rongrong Ji. 2022. X-CLIP: End-to-End Multi-grained Contrastive Learning for Video-Text Retrieval. In *Proceedings of the 30th ACM International Conference on Multimedia* (<conf-loc>, <city>Lisboa</city>, <country>Portugal</country>, </conf-loc>) *(MM '22)*. Association for Computing Machinery, New York, NY, USA, 638–647. https://doi.org/10.1145/3503161.3547910

Wendong Mao, Shuai Yang, Huihong Shi, Jiaying Liu, and Zhongfeng Wang. 2022. Intelligent typography: Artistic text style transfer for complex texture and structure. *IEEE Transactions on Multimedia* (2022).

Yifang Men, Zhouhui Lian, Yingmin Tang, and Jianguo Xiao. 2019. DynTypo: Example-Based Dynamic Text Effects Transfer. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* https://doi.org/10.1109/cvpr.2019.00602

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *Commun. ACM* 65, 1 (2021), 99–106.

Mitsuru Minakuchi and Yutaka Kidawara. 2008. Kinetic typography for ambient displays. In *Proceedings of the 2nd international conference on Ubiquitous information management and communication.* https://doi.org/10.1145/1352793.1352805

Mitsuru Minakuchi and Katsumi Tanaka. 2005. Automatic kinetic typography composer. In *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology.* https://doi.org/10.1145/1178477.1178512

Haomiao Ni, Changhao Shi, Kai Li, Sharon X Huang, and Martin Renqiang Min. 2023. Conditional Image-to-Video Generation with Latent Flow Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 18444–18455.

Hao Ouyang, Qiuyu Wang, Yuxi Xiao, Qingyan Bai, Juntao Zhang, Kecheng Zheng, Xiaowei Zhou, Qifeng Chen, and Yujun Shen. 2023. Codef: Content deformation fields for temporally consistent video processing. *arXiv preprint arXiv:2308.07926* (2023).

Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. 2021. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision.* 5865–5874.

Laurence Penny. 1996. *A History of TrueType.* Retrieved Mar 19, 2024 from https://www.truetype-typography.com

Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. 2023. DreamFusion: Text-to-3D using 2D Diffusion. In *The Eleventh International Conference on Learning Representations.* https://openreview.net/forum?id=FjNys5c7VyY

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

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Bjorn Ommer. 2022. High-Resolution Image Synthesis with Latent Diffusion Models. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* https://doi.org/10.1109/cvpr52688.2022.01042

Xiaoyu Shi, Zhaoyang Huang, Fu-Yun Wang, Weikang Bian, Dasong Li, Yi Zhang, Manyuan Zhang, Ka Chun Cheung, Simon See, Hongwei Qin, et al. 2024. Motion-I2V: Consistent and Controllable Image-to-Video Generation with Explicit Motion Modeling. *arXiv preprint arXiv:2401.15977* (2024).

Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. 2019. First Order Motion Model for Image Animation. *Neural Information Processing Systems,Neural Information Processing Systems* (Jan 2019).

Maham Tanveer, Yizhi Wang, Ali Mahdavi-Amiri, and Hao Zhang. 2023. DS-Fusion: Artistic Typography via Discriminated and Stylized Diffusion. (Mar 2023).

Zachary Teed and Jia Deng. 2020. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*. Springer, 402–419.

Jiuniu Wang, Hangjie Yuan, Dayou Chen, Yingya Zhang, Xiang Wang, and Shiwei Zhang. 2023. Modelscope text-to-video technical report. *arXiv preprint arXiv:2308.06571* (2023).

Wenjing Wang, Jiaying Liu, Shuai Yang, and Zongming Guo. 2019. Typography With Decor: Intelligent Text Style Transfer. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/cvpr.2019.00604

Xiang Wang, Hangjie Yuan, Shiwei Zhang, Dayou Chen, Jiuniu Wang, Yingya Zhang, Yujun Shen, Deli Zhao, and Jingren Zhou. 2024. Videocomposer: Compositional video synthesis with motion controllability. *Advances in Neural Information Processing Systems* 36 (2024).

Yizhi Wang and Zhouhui Lian. 2021. DeepVecFont: Synthesizing High-quality Vector Fonts via Dual-modality Learning. *ACM Transactions on Graphics* 40, 6 (2021), 15 pages. https://doi.org/10.1145/3478513.3480488

Liwenhan Xie, Zhaoyu Zhou, Kerun Yu, Yun Wang, Huamin Qu, and Siming Chen. 2023. Wakey-Wakey: Animate Text by Mimicking Characters in a GIF. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. https://doi.org/10.1145/3586183.3606813

Jinbo Xing, Menghan Xia, Yong Zhang, Haoxin Chen, Xintao Wang, Tien-Tsin Wong, and Ying Shan. 2023. Dynamicrafter: Animating open-domain images with video diffusion priors. *arXiv preprint arXiv:2310.12190* (2023).

Shuai Yang, Zhouhui Lian, and Zhongwen Guo. 2016. Awesome Typography: Statistics-Based Text Effects Transfer. *Cornell University - arXiv,Cornell University - arXiv* (Nov 2016).

Shuai Yang, Zhangyang Wang, and Jiaying Liu. 2021. Shape-Matching GAN++: Scale Controllable Dynamic Artistic Text Style Transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (Jan 2021), 1–1. https://doi.org/10.1109/tpami.2021.3055211

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 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.

Daquan Zhou, Weimin Wang, Hanshu Yan, Weiwei Lv, Yizhe Zhu, and Jiashi Feng. 2022. MagicVideo: Efficient Video Generation With Latent Diffusion Models. (Nov 2022).