Breathing Life Into Sketches Using Text-to-Video Priors

Rinon Gal*,1,2 Yael Vinker*,1 Yuval Alaluf[2] Amit Bermano[1]
Daniel Cohen-Or[1] Ariel Shamir[3] Gal Chechik[2]

[1]Tel-Aviv University [2]NVIDIA [3]Reichman University

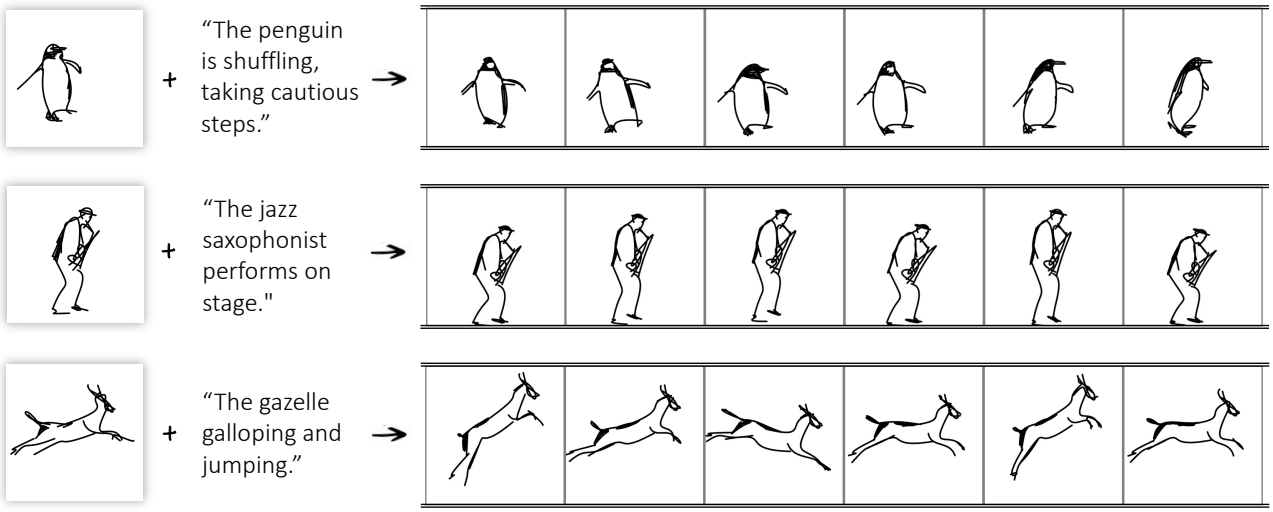


Figure 1. Given a still sketch in vector format and a text prompt describing a desired action, our method automatically animates the drawing with respect to the prompt. Please see the full animations in our project page: https://livesketch.github.io/

Abstract

A sketch is one of the most intuitive and versatile tools humans use to convey their ideas visually. An animated sketch opens another dimension to the expression of ideas and is widely used by designers for a variety of purposes. Animating sketches is a laborious process, requiring extensive experience and professional design skills. In this work, we present a method that automatically adds motion to a single-subject sketch (hence, "breathing life into it"), merely by providing a text prompt indicating the desired motion. The output is a short animation provided in vector representation, which can be easily edited. Our method does not require extensive training, but instead leverages the motion prior of a large pretrained text-to-video diffusion model using a score-distillation loss to guide the placement of strokes. To promote natural and smooth motion and to better preserve the sketch's appearance, we model the learned motion through two components. The first governs small local deformations and the second controls global affine transformations. Surprisingly, we find that even models that struggle to generate sketch videos on their own can still serve as a useful backbone for animating abstract representations.

*Indicates Equal Contribution. Order determined by coin flip.

1. Introduction

Sketches serve as a fundamental and intuitive tool for visual expression and communication [3, 20, 26]. Sketches capture the essence of visual entities with a few strokes, allowing humans to communicate abstract visual ideas. In this paper, we propose a method to "breathe life" into a static sketch by generating semantically meaningful short videos from it. Such animations can be useful for storytelling, illustrations, websites, presentations, and just for fun.

Animating sketches using conventional tools (such as Adobe Animate and Toon Boom) is challenging even for experienced designers [75], requiring specific artistic expertise. Hence, long-standing research efforts in computer graphics sought to develop automatic tools to simplify this process. However, these tools face multiple hurdles, such as a need to identify the semantic component of the sketch, or learning to create motion that appears natural. As such, existing methods commonly rely on user-annotated skeletal key points [17, 73] or user-provided reference motions that align with the sketch semantics [9, 75, 87].

In this work, we propose to bring a given static sketch to life, based on a textual prompt, without the need for any human annotations or explicit reference motions. We do so by leveraging a pretrained text-to-video diffusion model [43]. Several recent works propose using the prior of such mod-

els to bring life to a static *image* [63, 83, 93]. However, sketches pose distinct challenges, which existing methods fail to tackle as they are not designed with this domain in mind. Our method takes the recent advancement in text-to-video models into this new realm, aiming to tackle the challenging task of sketch animation. For this purpose, we propose specific design choices considering the delicate characteristics of this abstract domain.

In line with prior sketch generation approaches [79, 80], we use a vector representation of sketches, defining a sketch as a set of strokes (cubic Bézier curves) parameterized by their control points. Vector representations are popular among designers as they offer several advantages compared to pixel-based images. They are resolution-independent, *i.e*. can be scaled without losing quality. Moreover, they are easily editable: one can modify the sketch's appearance by choosing different stroke styles or change its shape by dragging control points. Additionally, their sparsity promotes smooth motion while preventing pixelization and blurring.

To bring a static sketch to life, we train a network to modify the stroke parameters for each video frame with respect to a given text prompt. Our method is optimization-based and requires no data or fine-tuning of large models. In addition, our method is general and can easily adapt to different text-to-video models, facilitating the use of future advancements in this field.

We train the network using a score-distillation sampling (SDS) loss [67]. This loss was designed to leverage pretrained text-to-*image* diffusion models for the optimization of non-pixel representations (e.g., NeRFs [55, 58] or SVGs [38, 39]) to meet given text-based constraints. We use this loss to extract motion priors from pretrained text-to-*video* diffusion models [32, 83]. Importantly, this allows us to inherit the internet-scale knowledge embedded in such models, enabling animation for a wide range of subjects across multiple categories.

We seperate the object movement into two components: local motion and global motion. Local motion aims to capture isolated, local effects (a saxophone player bending their knee). Conversely, global motion affects the object shape as a whole and is modeled through a per-frame transformation matrix. It can thus capture rigid motion (a penguin hobbling across the frame), or coordinate effects (the same penguin growing in size as it approaches the camera). We find that this separation is crucial in generating motion that is both locally smooth and globally significant while remaining faithful to the original characteristics of the subject.

We animate sketches from various domains and demonstrate the effectiveness of our approach in producing smooth and natural motion that conveys the intention of the control text while better preserving the shape and appearance of the input sketch.

We compare our results with recent pixel-based approaches highlighting the advantage of vector-based animation in the sketch domain. Our work allows anyone to breath life into their sketch in a simple and intuitive manner.

2. Previous Work

Sketches Free-hand sketching is a valuable tool for expressing ideas, concepts, and actions [20, 21, 30]. Extensive research has been conducted on the automatic generation of sketches [94]. Some works utilize pixel representation [40, 47, 74, 90], while others employ vector representation [6, 7, 13, 28, 50, 52, 56, 60, 70]. Several works propose a unified algorithm to produce sketches with a variety of styles [11, 53, 96] or at varying levels of abstraction [5, 61, 79, 80]. Traditional methods for sketch generation commonly rely on human-drawn sketch datasets. More recently, some works [22, 79, 80] incorporated the prior of large pretrained language-vision models to eliminate the dependency on such datasets. We also rely on such priors, and use a vector-based approach to depict our sketches, as it is a more natural representation for sketches and finds widespread use in character animation.

Sketch-based animation A long-standing area of interest in computer graphics aims to develop intuitive tools for creating life-like animations from still inputs. In character animation, motion is often represented as a temporal sequence of poses. These poses are commonly represented via user provided annotations, such as stick figures [14], skeletons [46, 65], or partial bone lines [64]. An alternative line of work represents motion through user provided 2D paths [15, 25, 36, 77], or through space-time curves [27]. However, these approaches still require some expertise and manual work to adjust different keyframes. Some methods assist animation by interactively predicting what users will draw next [1, 81, 91]. However, they still require manual sketching operations for each keyframe.

Rather than relying on user-created motion, some works propose to extract motion from real videos by statistical analysis of datasets [59], or by applying dynamic deformations extracted from a driving video [75]. Others turn to physically-based motion effects [41, 92], or learn to synthesize animations of hand-drawn 2D characters using a set of images depicting the character in various poses [17, 31, 68].

Drawing on 3D literature, some works aim to "wake up" a photo or a painting, extracting a textured human mesh from the image and moving it using pre-defined animations [33, 42, 86]. More recently, given a hand-drawn sketch of a human figure, Smith *et al*. [73] construct a character rig onto which they re-target motion capture data. Their approach is similarly limited to human figures and a predefined set of movements. Moreover, it commonly requires direct human intervention to fix skeleton joint estimations.

In contrast to these methods, our method requires only

a single sketch and no skeletons or explicit references. Instead, it leverages the strong prior of text-to-video generative models and generalizes across a wide range of animations described by free-form prompts.

Text-to-video generation Early works explored expanding the capabilities of recurrent neural networks [4, 10, 16], GANs [44, 51, 66, 78, 99], and auto-regressive transformers [85, 88, 89, 95] from image generation to video generation. However, these works primarily focused on generating videos within limited domains.

More recent research extends the capabilities of powerful text-to-image diffusion models to video generation by incorporating additional temporal attention modules into existing models or by temporally aligning an image decoder [8, 54, 72, 84]. Commonly, such alignment is performed in a latent space [2, 19, 49, 54, 82, 98]. Others train cascaded diffusion models [32], or learn to directly generate videos within a lower-dimensional 3D latent space [29].

We propose to extract the motion prior from such models and apply it to a vector sketch representation.

Image-to-video generation A closely related research area is image-to-video generation, where the goal is to animate an input image. Make-It-Move [34] train an encoder-decoder architecture to generate video sequences conditioned on an input image and a driving text prompt. Latent Motion Diffusion [35] learn the optical flow between pairs of video frames and use a 3D-UNet-based diffusion model to generate the resulting video sequence. CoDi [76] align multiple modalities (text, image, audio, and video) to a shared conditioning space and output space. ModelScope [82] train a latent video diffusion model, conditioned on an image input. Others first caption an image, then use the caption to condition a text-to-video model [54]. VideoCrafter [12] train a model conditioned on both text and image, with a special focus preserving the content, structure, and style of this image. Gen-2 [71] also operate in this domain, though their model's details are not public.

While showing impressive results in the pixel domain, these methods struggle to generalize to sketches. Our method is designed for sketches, constraining the outputs to vector representations that better preserve both the domain, and the characteristics of the input sketch.

3. Preliminaries

Vector representation Vector graphics allow us to create visual images directly from geometric shapes such as points, lines, curves, and polygons. Unlike raster images (represented with pixels), vector representation is resolution-free, more compact, and easier to modify. This quality makes vector images the preferred choice for various design applications, such as logo design, prints, animation, CAD, typography, web design, infographics, and illustrations. Scalable Vector Graphics (SVG) stands out as a popular vector image format due to its excellent support for interactivity and animation. We employ a differentiable rasterizer [48] to convert a vector image into its pixel-based image. This lets us manipulate the vector content using raster-based loss functions, as described below.

Score-Distillation Sampling The score-distillation sampling (SDS) loss, first proposed in Poole *et al*. [67], serves as a means for extracting a signal from a pretrained text-to-image diffusion model.

In their seminal work, Poole *et al*. propose to first use a parametric image synthesis model (e.g., a NeRF [57]) to generate an image x. This image is then noised to some intermediate diffusion time step t:

$$x_t = \alpha_t x + \sigma_t \epsilon, \tag{1}$$

where α_t, σ_t are parameters dependant on the noising schedule of the pretrained diffusion model, and $\epsilon \in \mathbb{N}(0, 1)$ is a noise sample.

The noised image is then passed through the diffusion model, conditioned on a text-prompt c describing some desired scene. The diffusion model's output, $\epsilon_\theta(x_t, t, c)$, is a prediction of the noise added to the image. The deviation of this prediction from the true noise, ϵ, can serve as a measure of the difference between the input image and one that better matches the prompt. This measure can then be used to approximate the gradients to the initial image synthesis model's parameters, ϕ, that would better align its outputs with the prompt. Specifically,

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

where $w(t)$ is a constant that depends on α_t. This optimization process is repeated, with the parametric model converging toward outputs that match the conditioning prompt.

In our work, we use this approach to extract the motion prior learned by a text-to-video diffusion model.

4. Method

Our method begins with two inputs: a user-provided static sketch in vector format, and a text prompt describing the desired motion. Our goal is to generate a short video, in the same vector format, which depicts the sketched subject acting in a manner consistent with the prompt. We therefore define three objectives that our approach should strive to meet: (1) the output video should match the text prompt, (2) the characteristics of the original sketch should be preserved, and (3) the generated motion should appear natural and smooth. Below, we outline the design choices we use to meet each of these objectives.

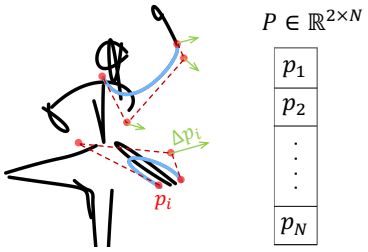


Figure 2. Data representation. Each curve (black or blue) is a cubic Bézier curve with 4 control points (red, shown for the blue curves). The total number of control points in the given sketch is denoted by N. For each frame and control point p_i, we learn a displacement Δp_i (green).

4.1. Representation

The input vector image is represented as a set of strokes placed over a white background, where each stroke is a two-dimensional Bézier curve with four control points. Each control point is represented by its coordinates: $p = (x, y) \in \mathbb{R}^2$. We denote the set of control points in a single frame with $P = \{p_1, ..p_N\} \in \mathbb{R}^{N\times 2}$, where N denotes the total number of points in the input sketch (see Figure 2). This number will remain fixed across all generated frames. We define a video with k frames as a sequence of k such sets of control points, and denote it by $Z = \{P^j\}_{j=1}^k \in \mathbb{R}^{N\cdot k\times 2}$.

Let P^{init} denote the set of points in the initial sketch. We duplicate P^{init} k times to create the initial set of frames Z^{init}. Our goal is to convert such a static sequence of frames into a sequence of frames animating the subject according to the motion described in the text prompt. We formulate this task as learning a set of 2D displacements $\Delta Z = \{\Delta p_i^j\}_{i\in N}^{j\in k}$, indicating the displacement of each point p_i^j, for each frame j (Fig. 2, in green).

4.2. Text-Driven Optimization

We begin by addressing our first objective: creating an output animation that aligns with the text prompt. We model the animation using a "neural displacement field" (Sec. 4.3), a small network $\mathcal{M}$ that receives as input the initial point set Z^{init} and predicts their displacements $\mathcal{M}(Z^{init}) = \Delta Z$. To train this network, we distill the motion prior encapsulated in a pretrained text-to-video diffusion model [83], using the SDS loss of Eq. (2).

At each training iteration (illustrated in Fig. 3), we add the predicted displacement vector ΔZ (marked in purple) to the initial set of points Z^{init} to form the sequence Z. We then use a differentiable rasterizer $\mathcal{R}$ [48], to transfer each set of per-frame points P^j to its corresponding frame in pixel space, denoted as $F^j = \mathcal{R}(P^j)$. The animated sketch is then defined by the concatenation of the rasterized frames, $F = \{F^1, ..F^k\} \in \mathbb{R}^{h\times w\times k}$.

Next, we sample a diffusion timestep t and noise $\epsilon \sim \mathcal{N}(0, 1)$. We use these to add noise to the rasterized video according to the diffusion schedule, creating F_t. This noisy video is then denoised using the pretrained text-to-video diffusion model ϵ_θ, where the diffusion model is conditioned on a prompt describing an animated scene (e.g., "a galloping horse"). Finally, we use Eq. (2) to update the parameters of $\mathcal{M}$ and repeat the process iteratively.

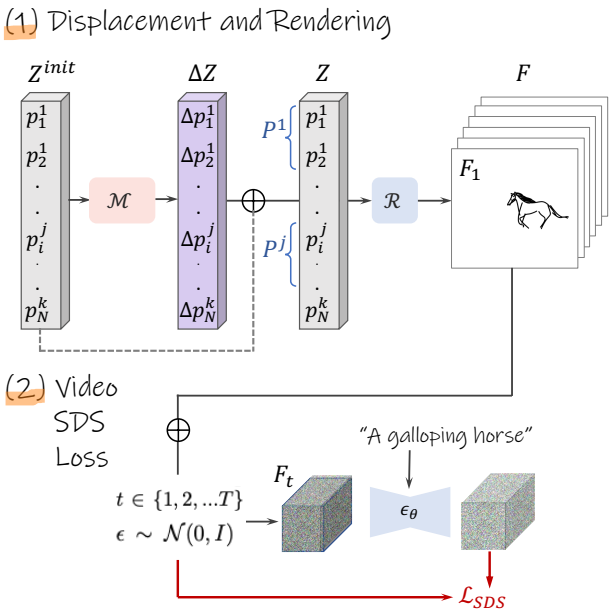


Figure 3. Text-driven optimization. At each training iteration: (1) We duplicate the initial control points across k frames and sum them with their predicted offsets. We render each frame and concatenate them to create the output video. (2) We use the SDS loss to extract a signal from a pretrained text-to-video model, which is used to update $\mathcal{M}$, the model that predicts the offsets.

The SDS loss thus guides $\mathcal{M}$ to learn displacements whose corresponding rasterized animation aligns with the desired text prompt. The extent of this alignment, and hence the intensity of the motion, is determined by optimization hyperparameters such as the diffusion guidance scale and learning rates. However, we find that increasing these parameters typically leads to artifacts such as jitter and shape-deformations, compromising both the fidelity of the original sketch and the fluidity of natural motion (see Sec. 5.2). As such, SDS alone fails to address our additional goals: (2) preserving the input sketch characteristics, and (3) creating natural motion. Instead, we tackle these goals through the design of our displacement field, $\mathcal{M}$.

4.3. Neural Displacement Field

We approach the network design with the intent of producing smoother motion with reduced shape deformations. We hypothesize that the artifacts observed with the unconstrained SDS optimization approach can be attributed in part to two mechanisms: (1) The SDS loss can be min-

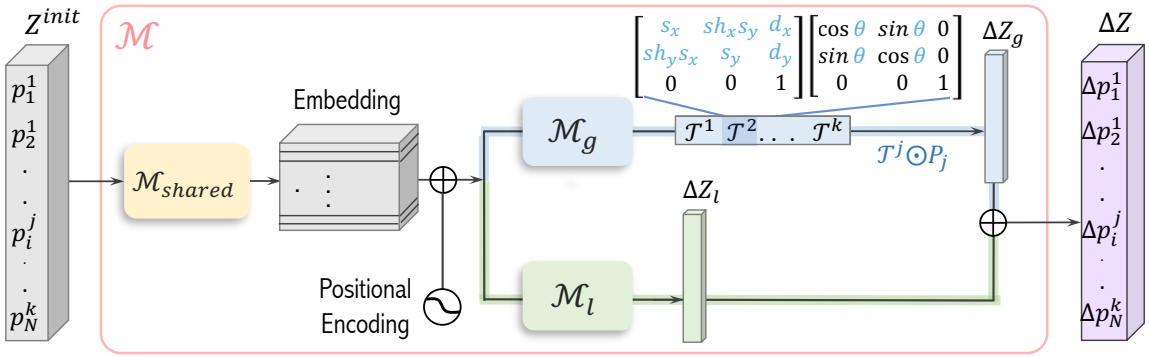


Figure 4. Network architecture. The input to the network is the initial set of control points Z^{init} (left, gray), and the output is the set of displacements ΔZ. The network consists of three parts. First, each control point p_i^j is projected with M_{shared} into a latent representation and summed with a positional encoding. These point features are passed to two different branches to predict global and local motion. The local motion predictor M_l (green) is a simple MLP that predicts an offset for each point (ΔZ_l), representing unconstrained local motion. The global motion predictor M_g predicts a per-frame transformation matrix $\mathcal{T}^j$ which applies scaling, shear, rotation, and translation. $\mathcal{T}^j$ is then applied to the points P_j in the corresponding frame to produce ΔZ_g. ΔZ is given by the sum: $\Delta Z = \Delta Z_g + \Delta Z_l$.

imized by deforming the generated shape into one that better aligns with the text-to-video model's semantic prior (e.g., prompting for a scuttling crab may lead to undesired changes in the shape of the crab itself). (2) Smooth motion requires small displacements at the local scale, and the network struggles to reconcile these with the large changes required for global translations. We propose to tackle both of these challenges by modeling our motion through two components: An unconstrained local-motion path, which models small deformations, and a global path which models affine transformations applied uniformly to an entire frame. This split will allow the network to separately model motion along both scales while restricting semantic changes in the path that controls greater scale movement. Below we outline the specific network design choices, as well as the parametrization that allows us to achieve this split.

Shared backbone Recall that our network, illustrated in Fig. 4, aims to map the initial control point set Z^{init} to their per-frame displacements $\mathcal{M}(Z^{init}) = \Delta Z$. Our first step is to create a shared feature backbone which will feed the separate motion paths. This component is built of an embedding step, where the coordinates of each control point are projected using a shared matrix $\mathcal{M}_{shared}$, and then summed with a positional encoding that depends on the frame index, and on the order of the point in the sketch. These point features are then fed into two parallel prediction paths: local, and global (Fig. 4, green and blue paths, respectively).

Local path The local path is parameterized by $\mathcal{M}_l$, a small MLP that takes the shared features and maps them to an offset ΔZ_l for every control point in Z^{init}. Here, the goal is to allow the network to learn unconstrained motion on its own to best match the given prompt. Indeed, in Sec. 5 we show that an unconstrained branch is crucial for the model to create meaningful motion. On the other hand, using this path to create displacements on the scale needed for global changes requires stronger SDS guidance or larger learning rates, leading to jitter and unwanted deformations at the local level. Hence, we delegate these changes to the global motion path. We note that similar behavior can be observed when directly optimizing the control points (*i.e.* without a network, following [38, 39], see Sec. 5.2).

Global path The goal of the global displacement prediction branch is to allow the model to capture meaningful global movements such as center-of-mass translation, rotation, or scaling, while maintaining the object's original shape. This path consists of a neural network, $\mathcal{M}_g$, that predicts a single global transformation matrix for each frame P^j. The matrix is then used to transform all control points of that frame, ensuring that the shape remains coherent. Specifically, we model the global motion as the sequential application of scaling, shear, rotation, and translation. These are parameterized using their standard affine matrix form (Fig. 4), which contains two parameters each for scale, shear, and translation, and one for rotation. Denoting the successive application of these transforms for frame j by $\mathcal{T}^j$, the global branch displacement for each point in this frame is then given by: $\Delta p^j_{i,global} = \mathcal{T}^j \odot p_i^{init} - p_i^{init}$.

We further extend the user's control over the individual components of the generated motion by adding a scaling parameter for each type of transformation: $\lambda_t, \lambda_r, \lambda_s$ and λ_{sh} for translation, rotation, scale, and shear, respectively. For example, let (d_x^j, d_y^j) denote the network's predicted translation parameters. We re-scale them as: $(d_x^j, d_y^j) \rightarrow (\lambda_t d_x^j, \lambda_t d_y^j)$. This allows us to attenuate specific aspects of motion that are undesired. For example, we can keep a subject roughly stationary by setting $\lambda_t = 0$. By modeling global changes through constrained transformations,

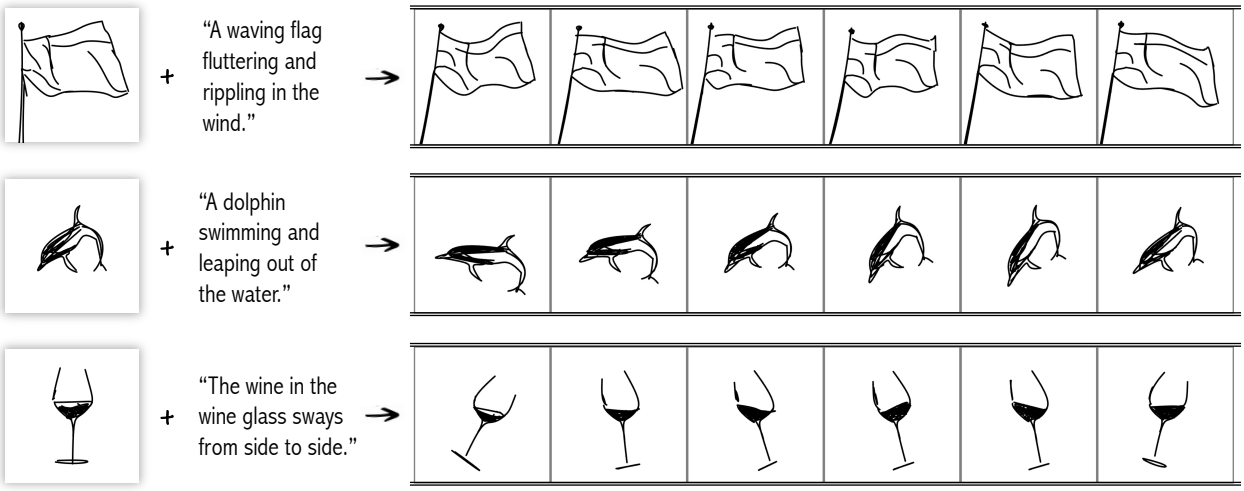


Figure 5. Qualitative results. Our model converts an initial sketch and a driving prompt describing some desired motion into a short video depicting the sketch moving according to the prompt. See the supplementary for the full videos and additional results.

applied uniformly to the entire frame, we limit the model's ability to create arbitrary deformations while preserving its ability to create large translations or coordinated effects.

Our final predicted displacements ΔZ are simply the sum of the two branches: $\Delta Z_l + \Delta Z_g$. The strength of these two terms (governed by the learning rates and guidance scales used to optimize each branch) will affect a tradeoff between our first goal (text-to-video alignment), and the other two goals (preserving the shape of the original sketch and creating smooth and natural motion). As such, a user can use this tradeoff to gain additional control over the generated video. For instance, prioritizing the preservation of sketch appearance by using a low learning rate for the local path, while affording greater freedom to global motion. We further demonstrate this tradeoff in the supplementary.

4.4. Training Details

We alternate between optimizing the local path and optimizing the global path. The shared backbone is optimized in both cases. Unless otherwise noted, we set the SDS guidance scale to 30 for the local path and 40 for the global path. We use Adam [45] with a learning rate of $1\mathrm{e}{-4}$ for the local path and a learning rate of $5\mathrm{e}{-3}$ for the global path. We find it useful to apply augmentations (random crops and perspective transformations) to the rendered videos during training. We further set $\lambda_t = 1.0, \lambda_r = 1\mathrm{e}{-2}, \lambda_s = 5\mathrm{e}{-2}, \lambda_{sh} = 1\mathrm{e}{-1}$. For our diffusion backbone, we use ModelScope text-to-video [82], but observe similar results with other backbones (see the supplementary file).

We optimize the networks for $1,000$ steps, taking roughly 30 minutes per video on a single A100 GPU. In practice, the model often converges after 500 steps (15 minutes). For additional training details, see the supplementary.

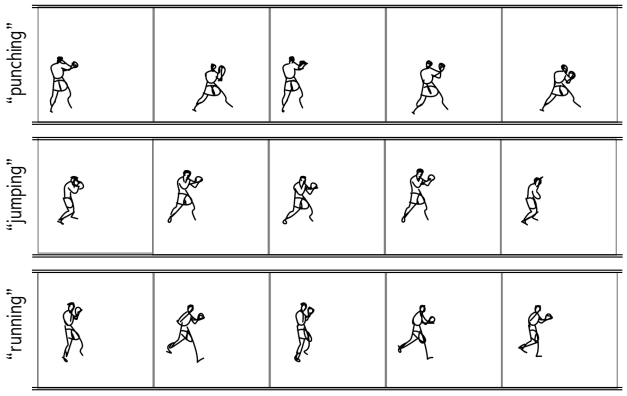


Figure 6. Our method can be used to animate the same sketch according to different prompts. These are typically restricted to actions that the portrayed subject would naturally perform. See the supplementary videos for more examples.

5. Results

We begin by showcasing our method's ability to animate a diverse set of sketches, following an array of text prompts (see Fig. 5 and supplementary videos). Our method can capture the delicate swaying of a dolphin in the water, follow a ballerina's dance routine, or mimic the gentle motion of wine swirling in a glass. Notably, it can apply these motions to sketches without any common skeleton or an explicit notation of parts. Moreover, our approach can animate the same sketch using different prompts (see Fig. 6), extending the freedom and diversity of text-to-video models to the more abstract sketch domain. Additional examples and full videos can be found in the supplementary materials.

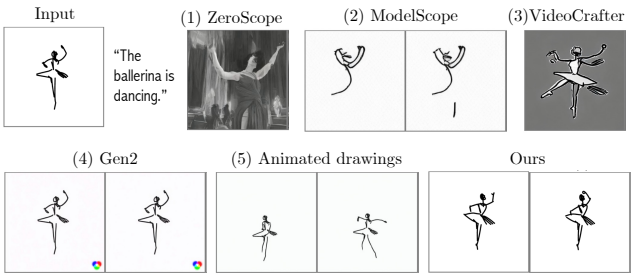


Figure 7. Qualitative comparisons. Image-to-video models suffer from artifacts and struggle to preserve the sketch shape (or even remain in a sketch domain). Animated drawings relies on skeletons and pre-captured reference motions. Hence, it cannot generalize to new domains. See the supplementary videos for more examples.

5.1. Comparisons

As no prior art directly tackles the reference-free sketch animation task, we explore two alternative approaches: pixel-based image-to-video approaches, and skeleton-based methods that build on pre-defined motions.

In the pixel-based scenario, we compare our method with four models: (1) ZeroScope image-to-video [54] which automatically captions the image [97] and uses the caption to prompt a text-to-video model. (2) ModelScope [82] image-to-video, which is directly conditioned on the image. (3) VideoCrafter [12] which is conditioned on both the image and the given text prompt, and (4) Gen-2 [71], a commercial web-based tool, conditioned on both image and text.

The results are shown in Fig. 7. We select representative frames from the output videos. The full videos are available in the supplementary material.

The results of ZeroScope and VideoCrafter show significant artifacts, and commonly fail to even produce a sketch. ModelScope fare batter, but struggle to preserve the shape of the sketch. Gen-2 either struggle to animate the sketch, or transforms it into a real image, depending on the input parameters (see the supplementary videos).

We further compare our approach with a skeleton and reference-based method [73] (Fig. 7, Animated Drawings). This method accounts for the sketch-based nature of our data and can better preserve its shape. However, it requires per-sketch manual annotations and is restricted to a predetermined set of human motions. Hence, it struggles to animate subjects which cannot be matched to a human skeleton, or whose motion does not align with the presets (see supplementary). In contrast, our method inherits the diversity of the text-to-video model and generalizes to multiple target classes without annotations or explicit references.

We additionally evaluate our method quantitatively. We compare with open methods that require no human intervention and can be evaluated at scale (ZeroScope [54], ModelScope [82], and Videocrafter [12]). We follow [79] and collect sketches spanning three categories: humans, animals, and objects. We asked ChatGPT to randomly select ten instances per category and suggest prompts describing their typical motion. We used CLIPasso [80] to generate a sketch for each subject. We applied our method and the alternative methods to these sketches and prompts, resulting in 30 animations per method (videos in the supplementary).

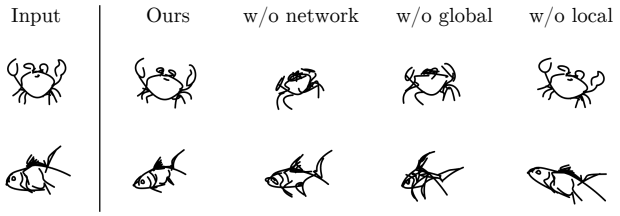


Figure 8. Qualitative ablation. Removing the neural network or the global path leads to shape deviations or jittery motion due to the need for higher learning rates (see supplementary videos). Modeling only global movement improves shape consistency, but fails to create realistic motion.

Following pixel-based methods [12, 19], we use CLIP [69] to measure the "sketch-to-video" consistency, defined as the average cosine similarity between the video's frames and the input sketch used to produce it.

We further evaluate the alignment between the generated videos and their corresponding prompts ("text-to-video alignment"). We use X-CLIP [62], a model that extends CLIP to video recognition. Here, we compare to the only baseline which is jointly guided by both image and text [12].

All results are provided in Tab. 1a. Our method outperforms the baselines on sketch-to-video consistency. In particular, it achieves significant gains over ModelScope whose text-to-video model serves as our prior. Moreover, our approach better aligns with the prompted motion, despite the use of a weaker text-to-video model as a backbone. These results, and in particular the ModelScope scores, demonstrate the importance of the vector representation which assists us in successfully extracting a motion prior without the low quality and artifacts introduced when trying to create sketches in the pixel domain.

5.2. Ablation Study

We further validate our suggested components through an ablation study. In particular, we evaluate the effect of using the neural prior in place of direct coordinate optimization and the effect of the global-local separation.

Qualitative results are shown in Fig. 8 (the corresponding videos are provided in the supplementary materials). As can be observed, removing the neural network can lead to increased jitter and harms shape preservation. Removing the global path leads to diminished movement across the frame and less coherent shape transformations. In contrast, removing the local path leads to unrealistic wobbling while keeping the original sketch almost unchanged.

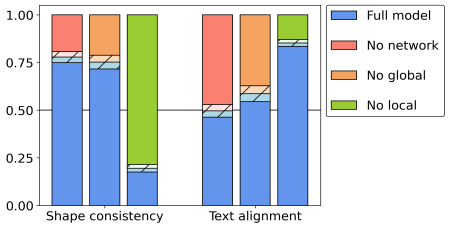
In Tab. 1b, we show quantitative results, following the

Method	Sketch-to-video consistency (↑)	Text-to-Video alignment(↑)
ZeroScope	0.754 ± 0.009	-
ModelScope	0.779 ± 0.009	-
VideoCrafter	0.876 ± 0.007	0.124 ± 0.005
Ours	$\mathbf{0.965} \pm 0.003$	$\mathbf{0.142} \pm 0.005$

(a) Comparisons to pixel-based approaches

Setup	Sketch-to-video consistency (↑)	Text-to-Video alignment(↑)
Full	0.965 ± 0.003	0.142 ± 0.005
No Net	0.926 ± 0.007	0.142 ± 0.005
No Glob.	0.936 ± 0.006	0.140 ± 0.005
No Local	0.970 ± 0.002	0.140 ± 0.004

(b) Ablation results



(c) User study

Table 1. Quantitative metrics. **(a)** CLIP-based consistency and text-video alignment comparisons to open-source image-to-video baselines. **(b)** The same CLIP-metrics used for an ablation study. **(c)** User study results. We pit our full model against each ablation setup. The blue bar indicates the percent of responders that preferred our full model over each baseline. Dashed area is one standard error.

same protocol as in Sec. 5.1. The sketch-to-video consistency results align with the qualitative observations. However, we observe that the metric for text-to-video alignment [62] is not sensitive enough to gauge the difference between our ablation setups (standard errors are larger than the gaps).

We additionally conduct a user study, based on a two-alternative forced-choice setup. Each user is shown two videos (one output from the full method, and one from a random ablation setup) and asked to select: (1) the video that better preserves the appearance of the initial sketch, and (2) the video that better matches the motion outlined in the prompt. We collected responses from 31 participants over 30 pairs. The results are provided in Tab. 1c.

Users considered the full method's text-to-video alignment to be on-par or better than all ablation setups. When considering sketch-to-video consistency, our method is preferred over both setups that create reasonable motion (no network and no global). Removing the local path leads to higher consistency with respect to the original frame, largely because the sketch remains almost unchanged. Our full method allows for more expressive motion, while still showing remarkable preservation of the input sketch.

In the supplementary materials, we provide further analysis on the effects of our hyperparameter choices, and highlight an emergent trade-off between shape preservation and the quality of generated motion.

6. Limitations

While our work enables sketch-animation across various classes and prompts, it comes with limitations. First, we build upon the sketch representation from [80]. However, sketches can be represented in many forms with different types of curves and primitive shapes. Using our method with other sketch representations could result in performance degradation. For instance, in Fig. 9(1) the surfer's scale has significantly changed. Addressing the diversity of vector sketches requires further development. Second, our method assumes a single-subject input sketch (a common scenario in character animation techniques). When applied to scene sketches or sketches with multiple objects, we observe reduced result quality due to the design constraints.

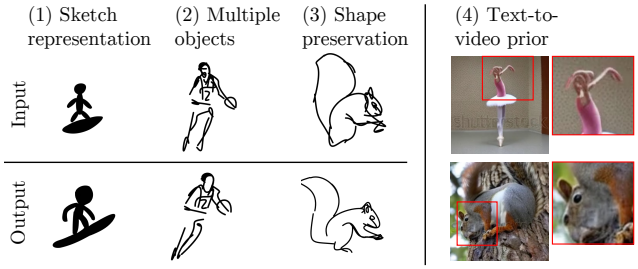


Figure 9. Method limitations. The method may struggle with certain sketch representations, fail to tackle multiple objects or complex scenes, or create undesired shape changes. Moreover, it is restricted to motions which the text-to-video prior can create.

For example in Fig. 9(2), the basketball cannot be separated from the player, contrary to the natural motion of dribbling.

Third, our method faces a trade-off between motion quality and sketch fidelity, and a diligent balance should be achieved between the two. In Fig. 9(3), the animated squirrel's appearance differs from the input sketch. This trade-off is further discussed in the supplementary material. Potential improvement lies in adopting a mesh-based representation with an approximate rigidity loss [37], or by trying to enforce consistency in the diffusion feature space [24] .

Finally, our approach inherits the limitations of text-to-video priors. Such models are trained on large-scale data, but may be unaware of specific motions, or portray strong biases. For example, as demonstrated in Fig. 9, the model we utilize tend to produce significant artefacts when used for text-to-video generation. However, our method is agnostic to the backbone model and hence could likely by used with newer, improved models as they become available, or with personalized models [23] that were augmented with new, unobserved motions.

7. Conclusions

We presented a technique to breath life into a given static sketch, following a text prompt. Our method builds on the motion prior captured by powerful text-to-video models. We show that even though these models struggle with generating sketches directly, they can still comprehend such

abstract representations in a semantically meaningful way, creating smooth and appealing motions. We hope that our work will facilitate further research to provide intuitive and practical tools for sketch animation that incorporate recent advances in text-based video generation.

8. Acknowledgements

We thank Oren Katzir and Guy Tevet for providing feedback on early versions of this manuscript. This work was partially supported by BSF (grant 2020280) and ISF (grants 2492/20 and 3441/21).

References

[1] Aseem Agarwala, Aaron Hertzmann, David Salesin, and Steven Seitz. Keyframe-based tracking of rotoscoping and animation. *ACM Trans. Graph.*, 23:584–591, 2004. 2

[2] Jie An, Songyang Zhang, Harry Yang, Sonal Gupta, Jia-Bin Huang, Jiebo Luo, and Xi Yin. Latent-shift: Latent diffusion with temporal shift for efficient text-to-video generation. *arXiv preprint arXiv:2304.08477*, 2023. 3

[3] Maxime Aubert, Adam Brumm, Muhammad Ramli, Thomas Sutikna, E Wahyu Saptomo, Budianto Hakim, Michael J Morwood, Gerrit D van den Bergh, Leslie Kinsley, and Anthony Dosseto. Pleistocene cave art from sulawesi, indonesia. *Nature*, 514(7521):223–227, 2014. 1

[4] Mohammad Babaeizadeh, Chelsea Finn, Dumitru Erhan, Roy H Campbell, and Sergey Levine. Stochastic variational video prediction. *arXiv preprint arXiv:1710.11252*, 2017. 3

[5] Itamar Berger, Ariel Shamir, Moshe Mahler, Elizabeth Carter, and Jessica Hodgins. Style and abstraction in portrait sketching. *ACM Trans. Graph.*, 32(4), 2013. 2

[6] Ayan Kumar Bhunia, Ayan Das, Umar Riaz Muhammad, Yongxin Yang, Timothy M. Hospedales, Tao Xiang, Yulia Gryaditskaya, and Yi-Zhe Song. Pixelor: a competitive sketching ai agent. so you think you can sketch? *ACM Trans. Graph.*, 39:166:1–166:15, 2020. 2

[7] Ankan Kumar Bhunia, Salman Khan, Hisham Cholakkal, Rao Muhammad Anwer, Fahad Shahbaz Khan, Jorma Laaksonen, and Michael Felsberg. Doodleformer: Creative sketch drawing with transformers. *ECCV*, 2022. 2

[8] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 3

[9] Christoph Bregler, Lorie Loeb, Erika Chuang, and Hrishi Deshpande. Turning to the masters: Motion capturing cartoons. *ACM Transactions on Graphics (TOG)*, 21(3):399–407, 2002. 1

[10] Lluis Castrejon, Nicolas Ballas, and Aaron Courville. Improved conditional vrnns for video prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7608–7617, 2019. 3

[11] Caroline Chan, Frédo Durand, and Phillip Isola. Learning to generate line drawings that convey geometry and semantics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7915–7925, 2022. 2

[12] Haoxin Chen, Menghan Xia, Yingqing He, Yong Zhang, Xiaodong Cun, Shaoshu Yang, Jinbo Xing, Yaofang Liu, Qifeng Chen, Xintao Wang, Chao Weng, and Ying Shan. Videocrafter1: Open diffusion models for high-quality video generation, 2023. 3, 7

[13] Yajing Chen, Shikui Tu, Yuqi Yi, and Lei Xu. Sketch-pix2seq: a model to generate sketches of multiple categories. *ArXiv*, abs/1709.04121, 2017. 2

[14] James Davis, Maneesh Agrawala, Erika Chuang, Zoran Popović, and David Salesin. A sketching interface for articulated figure animation. In *Acm siggraph 2006 courses*, pages 15–es. 2006. 2

[15] Richard C. Davis, Brien Colwell, and James A. Landay. K-sketch: A 'kinetic' sketch pad for novice animators. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, page 413–422, New York, NY, USA, 2008. Association for Computing Machinery. 2

[16] Emily Denton and Rob Fergus. Stochastic video generation with a learned prior. In *International conference on machine learning*, pages 1174–1183. PMLR, 2018. 3

[17] Marek Dvorožnák, Wilmot Li, Vladimir G Kim, and Daniel Sỳkora. Toonsynth: example-based synthesis of hand-colored cartoon animations. *ACM Transactions on Graphics (TOG)*, 37(4):1–11, 2018. 1, 2

[18] Mathias Eitz, James Hays, and Marc Alexa. How do humans sketch objects? *ACM Trans. Graph. (Proc. SIGGRAPH)*, 31 (4):44:1–44:10, 2012. 2

[19] Patrick Esser, Johnathan Chiu, Parmida Atighehchian, Jonathan Granskog, and Anastasis Germanidis. Structure and content-guided video synthesis with diffusion models. *arXiv preprint arXiv:2302.03011*, 2023. 3, 7

[20] Judy Fan, Wilma A. Bainbridge, Rebecca Chamberlain, and Jeffrey D. Wammes. Drawing as a versatile cognitive tool. *Nature Reviews Psychology*, 2:556 – 568, 2023. 1, 2

[21] Judith E. Fan, Daniel L. K. Yamins, and Nicholas B. Turk-Browne. Common object representations for visual production and recognition. *Cognitive science*, 42 8:2670–2698, 2018. 2

[22] Kevin Frans, Lisa B. Soros, and Olaf Witkowski. Clipdraw: Exploring text-to-drawing synthesis through language-image encoders. *CoRR*, abs/2106.14843, 2021. 2, 3

[23] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit Haim Bermano, Gal Chechik, and Daniel Cohen-or. An image is worth one word: Personalizing text-to-image generation using textual inversion. In *The Eleventh International Conference on Learning Representations*, 2022. 8

[24] Michal Geyer, Omer Bar-Tal, Shai Bagon, and Tali Dekel. Tokenflow: Consistent diffusion features for consistent video editing. *arXiv preprint arxiv:2307.10373*, 2023. 8

[25] Michael Gleicher. Motion path editing. In *Proceedings of the 2001 Symposium on Interactive 3D Graphics*, page 195–202, New York, NY, USA, 2001. Association for Computing Machinery. 2

[26] Ernst Hans Gombrich. *The story of art*. Phaidon London, 1995. 1

[27] Martin Guay, Rémi Ronfard, Michael Gleicher, and Marie-Paule Cani. Space-time sketching of character animation. *ACM Transactions on Graphics (ToG)*, 34(4):1–10, 2015. 2

[28] David Ha and Douglas Eck. A neural representation of sketch drawings. *CoRR*, abs/1704.03477, 2017. 2, 3

[29] Yingqing He, Tianyu Yang, Yong Zhang, Ying Shan, and Qifeng Chen. Latent video diffusion models for high-fidelity long video generation. 2022. 3

[30] Aaron Hertzmann. Why do line drawings work? a realism hypothesis. *Perception*, 49:439 – 451, 2020. 2

[31] Tobias Hinz, Matthew Fisher, Oliver Wang, Eli Shechtman, and Stefan Wermter. Charactergan: Few-shot keypoint character animation and reposing. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1988–1997, 2022. 2

[32] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022. 2, 3

[33] Alexander Hornung, Ellen Dekkers, and Leif Kobbelt. Character animation from 2d pictures and 3d motion data. *ACM Trans. Graph.*, 26(1):1–es, 2007. 2

[34] Yaosi Hu, Chong Luo, and Zhenzhong Chen. Make it move: controllable image-to-video generation with text descriptions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18219–18228, 2022. 3

[35] Yaosi Hu, Zhenzhong Chen, and Chong Luo. Lamd: Latent motion diffusion for video generation, 2023. 3

[36] Takeo Igarashi, Rieko Kadobayashi, Kenji Mase, and Hidehiko Tanaka. Path drawing for 3d walkthrough. In *Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology*, page 173–174, New York, NY, USA, 1998. Association for Computing Machinery. 2

[37] Takeo Igarashi, Tomer Moscovich, and John F. Hughes. As-rigid-as-possible shape manipulation. *ACM Trans. Graph.*, 24(3):1134–1141, 2005. 8

[38] Shir Iluz, Yael Vinker, Amir Hertz, Daniel Berio, Daniel Cohen-Or, and Ariel Shamir. Word-as-image for semantic typography. *ACM Trans. Graph.*, 42(4), 2023. 2, 5

[39] Ajay Jain, Amber Xie, and Pieter Abbeel. Vectorfusion: Text-to-svg by abstracting pixel-based diffusion models. *arXiv*, 2022. 2, 5, 3

[40] Moritz Kampelmühler and Axel Pinz. Synthesizing human-like sketches from natural images using a conditional convolutional decoder. *CoRR*, abs/2003.07101, 2020. 2

[41] Rubaiat Kazi, Fanny Chevalier, Tovi Grossman, Shengdong Zhao, and George Fitzmaurice. Draco: Bringing life to illustrations with kinetic textures. *Conference on Human Factors in Computing Systems - Proceedings*, 2014. 2

[42] Levon Khachatryan. Tex-an mesh: Textured and animatable human body mesh reconstruction from a single image. https://github.com/lev1khachatryan/Tex-An_Mesh, 2020. 2

[43] Levon Khachatryan, Andranik Movsisyan, Vahram Tadevosyan, Roberto Henschel, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. Text2video-zero: Text-to-image diffusion models are zero-shot video generators. *arXiv preprint arXiv:2303.13439*, 2023. 1

[44] Doyeon Kim, Donggyu Joo, and Junmo Kim. Tivgan: Text to image to video generation with step-by-step evolutionary generator. *IEEE Access*, 8:153113–153122, 2020. 3

[45] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[46] Zohar Levi and Craig Gotsman. ArtiSketch: A System for Articulated Sketch Modeling. *Computer Graphics Forum*, 2013. 2

[47] Mengtian Li, Zhe Lin, Radomir Mech, Ersin Yumer, and Deva Ramanan. Photo-sketching: Inferring contour drawings from images, 2019. 2

[48] Tzu-Mao Li, Michal Lukáč, Gharbi Michaël, and Jonathan Ragan-Kelley. Differentiable vector graphics rasterization for editing and learning. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, 39(6):193:1–193:15, 2020. 3, 4

[49] Xin Li, Wenqing Chu, Ye Wu, Weihang Yuan, Fanglong Liu, Qi Zhang, Fu Li, Haocheng Feng, Errui Ding, and Jingdong Wang. Videogen: A reference-guided latent diffusion approach for high definition text-to-video generation. *arXiv preprint arXiv:2309.00398*, 2023. 3

[50] Yi Li, Yi-Zhe Song, Timothy M. Hospedales, and Shaogang Gong. Free-hand sketch synthesis with deformable stroke models. *CoRR*, abs/1510.02644, 2015. 2

[51] Yitong Li, Martin Min, Dinghan Shen, David Carlson, and Lawrence Carin. Video generation from text. In *Proceedings of the AAAI conference on artificial intelligence*, 2018. 3

[52] Hangyu Lin, Yanwei Fu, Yu-Gang Jiang, and X. Xue. Sketch-bert: Learning sketch bidirectional encoder representation from transformers by self-supervised learning of sketch gestalt. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6757–6766, 2020. 2

[53] Difan Liu, Matthew Fisher, Aaron Hertzmann, and Evangelos Kalogerakis. Neural strokes: Stylized line drawing of 3d shapes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14204–14213, 2021. 2

[54] Zhengxiong Luo, Dayou Chen, Yingya Zhang, Yan Huang, Liang Wang, Yujun Shen, Deli Zhao, Jingren Zhou, and Tieniu Tan. Videofusion: Decomposed diffusion models for high-quality video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 3, 7

[55] Gal Metzer, Elad Richardson, Or Patashnik, Raja Giryes, and Daniel Cohen-Or. Latent-nerf for shape-guided generation of 3d shapes and textures. *arXiv preprint arXiv:2211.07600*, 2022. 2

[56] Daniela Mihai and Jonathon Hare. Learning to draw: Emergent communication through sketching. *Advances in Neural Information Processing Systems*, 34:7153–7166, 2021. 2

[57] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf:

Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, pages 405–421. Springer, 2020. 3

[58] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 2

[59] Jianyuan Min, Yen-Lin Chen, and Jinxiang Chai. Interactive generation of human animation with deformable motion models. *ACM Trans. Graph.*, 29(1), 2009. 2

[60] Haoran Mo, Edgar Simo-Serra, Chengying Gao, Changqing Zou, and Ruomei Wang. General virtual sketching framework for vector line art. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2021)*, 40(4):51:1–51:14, 2021. 2

[61] Umar Riaz Muhammad, Yongxin Yang, Yi-Zhe Song, Tao Xiang, and Timothy M. Hospedales. Learning deep sketch abstraction. *CoRR*, abs/1804.04804, 2018. 2

[62] Bolin Ni, Houwen Peng, Minghao Chen, Songyang Zhang, Gaofeng Meng, Jianlong Fu, Shiming Xiang, and Haibin Ling. Expanding language-image pretrained models for general video recognition. 2022. 7, 8

[63] Haomiao Ni, Changhao Shi, Kai Li, Sharon X. Huang, and Martin Renqiang Min. Conditional image-to-video generation with latent flow diffusion models, 2023. 2

[64] A. Cengiz Öztireli, Ilya Baran, Tiberiu Popa, Boris Dalstein, Robert W. Sumner, and Markus Gross. Differential blending for expressive sketch-based posing. In *Proceedings of the 2013 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, New York, NY, USA, 2013. ACM. 2

[65] Junjun Pan and Jian J. Zhang. *Sketch-Based Skeleton-Driven 2D Animation and Motion Capture*, pages 164–181. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. 2

[66] Yingwei Pan, Zhaofan Qiu, Ting Yao, Houqiang Li, and Tao Mei. To create what you tell: Generating videos from captions. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1789–1798, 2017. 3

[67] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *The Eleventh International Conference on Learning Representations*, 2022. 2, 3

[68] Omid Poursaeed, Vladimir Kim, Eli Shechtman, Jun Saito, and Serge Belongie. Neural puppet: Generative layered cartoon characters. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3346–3356, 2020. 2

[69] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 7

[70] Leo Sampaio Ferraz Ribeiro, Tu Bui, John P. Collomosse, and Moacir Antonelli Ponti. Sketchformer: Transformer-based representation for sketched structure. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14141–14150, 2020. 2

[71] Runway. Gen-2: Text driven video generation. https://research.runwayml.com/gen2, 2023. 3, 7

[72] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022. 3

[73] Harrison Jesse Smith, Qingyuan Zheng, Yifei Li, Somya Jain, and Jessica K Hodgins. A method for animating children's drawings of the human figure. *ACM Transactions on Graphics*, 42(3):1–15, 2023. 1, 2, 7

[74] Jifei Song, Kaiyue Pang, Yi-Zhe Song, Tao Xiang, and Timothy Hospedales. Learning to sketch with shortcut cycle consistency, 2018. 2

[75] Qingkun Su, Xue Bai, Hongbo Fu, Chiew-Lan Tai, and Jue Wang. Live sketch: Video-driven dynamic deformation of static drawings. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2018. 1, 2

[76] Zineng Tang, Ziyi Yang, Chenguang Zhu, Michael Zeng, and Mohit Bansal. Any-to-any generation via composable diffusion. 2023. 3

[77] Matthew Thorne, David Burke, and Michiel Van De Panne. Motion doodles: an interface for sketching character motion. *ACM Transactions on Graphics (ToG)*, 23(3):424–431, 2004. 2

[78] Yu Tian, Jian Ren, Menglei Chai, Kyle Olszewski, Xi Peng, Dimitris N. Metaxas, and Sergey Tulyakov. A good image generator is what you need for high-resolution video synthesis. In *International Conference on Learning Representations*, 2021. 3

[79] Yael Vinker, Yuval Alaluf, Daniel Cohen-Or, and Ariel Shamir. Clipascene: Scene sketching with different types and levels of abstraction. 2022. 2, 7, 3

[80] Yael Vinker, Ehsan Pajouheshgar, Jessica Y. Bo, Roman Christian Bachmann, Amit Haim Bermano, Daniel Cohen-Or, Amir Zamir, and Ariel Shamir. Clipasso: Semantically-aware object sketching. *ACM Trans. Graph.*, 41(4), 2022. 2, 7, 8, 3

[81] Jue Wang, Yingqing Xu, Heung-Yeung Shum, and Michael F. Cohen. Video tooning. In *ACM SIGGRAPH 2004 Papers*, page 574–583, New York, NY, USA, 2004. Association for Computing Machinery. 2

[82] Jiuniu Wang, Hangjie Yuan, Dayou Chen, Yingya Zhang, Xiang Wang, and Shiwei Zhang. Modelscope text-to-video technical report. *arXiv preprint arXiv:2308.06571*, 2023. 3, 6, 7

[83] Xiang* Wang, Hangjie* Yuan, Shiwei* Zhang, Dayou* Chen, Jiuniu Wang, Yingya Zhang, Yujun Shen, Deli Zhao, and Jingren Zhou. Videocomposer: Compositional video synthesis with motion controllability. *arXiv preprint arXiv:2306.02018*, 2023. 2, 4

[84] Yaohui Wang, Xinyuan Chen, Xin Ma, Shangchen Zhou, Ziqi Huang, Yi Wang, Ceyuan Yang, Yinan He, Jiashuo Yu, Peiqing Yang, et al. Lavie: High-quality video generation with cascaded latent diffusion models. *arXiv preprint arXiv:2309.15103*, 2023. 3

[85] Dirk Weissenborn, Oscar Täckström, and Jakob Uszkoreit. Scaling autoregressive video models. *arXiv preprint arXiv:1906.02634*, 2019. 3

[86] Chung-Yi Weng, Brian Curless, and Ira Kemelmacher-Shlizerman. Photo wake-up: 3d character animation from a single photo. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5901–5910, 2018. 2

[87] Chung-Yi Weng, Brian Curless, and Ira Kemelmacher-Shlizerman. Photo wake-up: 3d character animation from a single photo. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5908–5917, 2019. 1

[88] Chenfei Wu, Lun Huang, Qianxi Zhang, Binyang Li, Lei Ji, Fan Yang, Guillermo Sapiro, and Nan Duan. Godiva: Generating open-domain videos from natural descriptions. *arXiv preprint arXiv:2104.14806*, 2021. 3

[89] Chenfei Wu, Jian Liang, Lei Ji, Fan Yang, Yuejian Fang, Daxin Jiang, and Nan Duan. Nüwa: Visual synthesis pre-training for neural visual world creation. In *European conference on computer vision*, pages 720–736. Springer, 2022. 3

[90] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *Proceedings of the IEEE international conference on computer vision*, pages 1395–1403, 2015. 2

[91] Jun Xing, Li-Yi Wei, Takaaki Shiratori, and Koji Yatani. Autocomplete hand-drawn animations. *ACM Trans. Graph.*, 34 (6), 2015. 2

[92] Jun Xing, Rubaiat Kazi, Tovi Grossman, Li-Yi Wei, Jos Stam, and George Fitzmaurice. Energy-brushes: Interactive tools for illustrating stylized elemental dynamics. pages 755–766, 2016. 2

[93] Jinbo Xing, Menghan Xia, Yong Zhang, Haoxin Chen, Xintao Wang, Tien-Tsin Wong, and Ying Shan. Dynamicrafter: Animating open-domain images with video diffusion priors. *arXiv preprint arXiv:2310.12190*, 2023. 2

[94] Peng Xu, Timothy M. Hospedales, Qiyue Yin, Yi-Zhe Song, Tao Xiang, and Liang Wang. Deep learning for free-hand sketch: A survey and a toolbox, 2020. 2

[95] Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. Videogpt: Video generation using vq-vae and transformers. *arXiv preprint arXiv:2104.10157*, 2021. 3

[96] Ran Yi, Yong-Jin Liu, Yu-Kun Lai, and Paul L Rosin. Unpaired portrait drawing generation via asymmetric cycle mapping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8217–8225, 2020. 2

[97] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. 2022. 7

[98] Daquan Zhou, Weimin Wang, Hanshu Yan, Weiwei Lv, Yizhe Zhu, and Jiashi Feng. Magicvideo: Efficient video generation with latent diffusion models, 2023. 3

[99] Jingyuan Zhu, Huimin Ma, Jiansheng Chen, and Jian Yuan. Motionvideogan: A novel video generator based on the motion space learned from image pairs. *IEEE Transactions on Multimedia*, 2023. 3

Breathing Life Into Sketches Using Text-to-Video Priors

Supplementary Material

Table of Contents

A. Additional results and videos

All videos and a large number of additional results are available in our supplementary website. These include an array of subjects animated with our method, along with additional comparisons, ablation experiments and visualizations of limitations. Please note that all comparisons and ablation baseline results use our default parameters, while the large video gallery includes results with different parameter settings, chosen according to our aesthetic preferences.

B. Analysis and ablation

In this section we present an array of experiments that explore the sensitivity of our method to different hyperparameters of the approach. These include technical changes (such as learning rate adjustments), but also conceptual explorations such as the effect of sketch abstraction on the generated videos.

B.1. Text prompt effect

Our animation process is guided by a user-provided text, based on the prior of a pretrained text-to-video model. This section further examines how the specified prompt affects the animation. We first verify that the text itself influences the results in a meaningful way. To do so, we apply our method to several example sketches, using two alternatives: A "generic" prompt ("the object is moving"), and the empty prompt (""). The results are shown in Fig. 10 and in the "Text Prompt Effect" section of the website. Using the generic prompt leads to irrelevant animations in which both

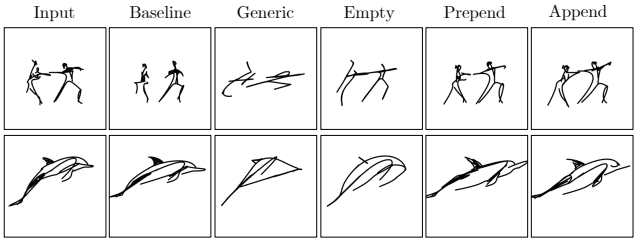


Figure 10. Text prompt effect. We investigate the effects of using a generic prompt ("The object is moving") for all sketches, the effect of using an empty prompt, or prepending and appending strings that compel the diffusion model go generate sketches. Additional video results are shown in the website.

the motion and the sketch appearance exhibit significant artifacts. Using an empty prompt leads to results with no visible motion, and large shape deviations. We can thus conclude that using prompts tailored for the input sketch is crucial, both to preserve its characteristics and for the ability to generate meaningful motion.

We further examine the impact of modifying the prompt in a way that would motivate the text-to-video to create a sketch. Specifically, we either prepend the string "A sketch of" or append the string "Abstract sketch. Line drawing" to the prompts.

In general, explicitly prompting for a sketch works comparably well to the original prompts. In some cases we observe slight differences in the extent of the motion or in the adherence to slight details in the input sketch (*e.g.* the penguin's left fin is filled out when using the sketch prompts). However, these can likely be accounted for with learning rate tuning. We thus conclude that the model can reasonably infer the semantics of the object even when the prompt does not directly convey its sketch-based nature.

Finally, we show additional results for applying different prompts to the same input sketch (see "Varying the Prompt" in the provided website). For example, observe how the boxer changes his motion in accordance with the texts provided, demonstrating the actions of jumping, running, and punching. Similarly, a cat can be made to change its pose, or walk towards the camera. However, in some cases the method is not sensitive enough to the changes in the provided text prompt. This is particularly apparent when the prompt requests large changes in the shape of the subject, or when the diffusion model struggles to generate the described motion even in it's basic text-to-video setup. In the video website, we demonstrate this on the ballerina sketch, where the specifics of the prompts are largely ignored, lead-

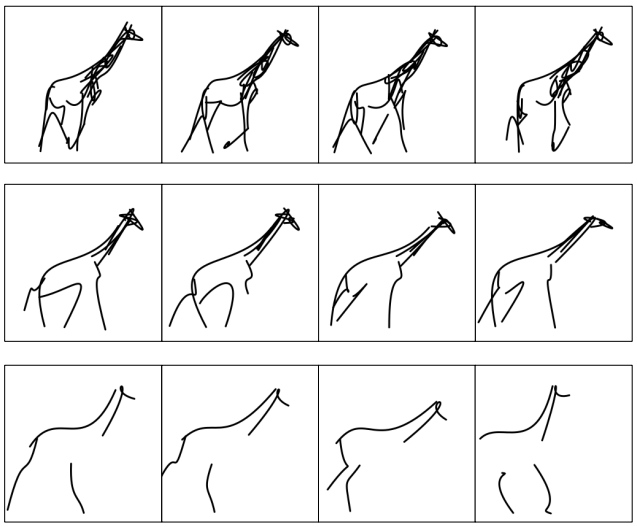


Figure 11. Different levels of abstraction. We show four selected frames for each level of abstraction. The model can successfully synthesize movement even for very abstract representations.

ing to similar dancing motions. However, notice that supplying the base diffusion model with those same prompts, also creates videos with dancing that is unrelated to the motion described in the prompt. We hope that this limitation could be overcome as better, more expressive text-to-video models become available.

B.2. Different levels of abstraction

We also demonstrate the effect of altering the abstraction level of the input sketches. We show results for three objects with three levels of abstraction. The sketches were generated using 16, 8, and 4 strokes. An example is provided in Fig. 11, and more examples and the full videos are provided in the supplementary website's "Abstraction Level" section. As can be seen, even for the extreme case of very abstract sketches with only four strokes, our method still manages to produce animations that fit the given prompt. Yet, the abstract animations may appear less smooth, leaving room for future work to tackle such challenging cases.

B.3. Sketch representation

As described in the main paper, we represent a sketch as a set of black cubic Bezier curves, and use CLIPasso [80] to automatically generate the sketches shown in the paper. However, our approach can be applied to alternative sketch representations. As highlighted in the limitations section of the main paper, employing different sketch representations may require additional hyperparameter tuning. To illustrate the impact of changing the sketch representation, we applied our method to sketches from the TU-Berlin sketch dataset [18], a human-drawn class-based sketch dataset. We showcase the results of four representative sketches. Our

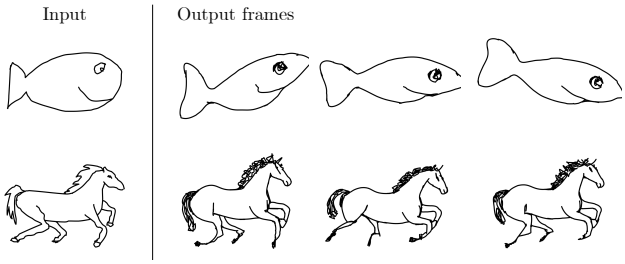


Figure 12. Human-drawn sketches. We applied our method to sketches from the TU-Berlin dataset. With our default parameters, these create reasonable motion but fail to preserve the exact sketch appearance. By tuning the parameters for this input style, shape preservation can be improved. See the website for examples.

method was directly applied to the provided SVG files. Fig. 12 shows a few representative frames from the videos produced for two sketches. More results are shown in the supplementary website. As can be seen, our method successfully animated the sketches, however their appearance is not fully preserved when using the default hyperparameters. This can be improved by using lower learning rates for the local path.

B.4. Learning rate scaling and tradeoffs

As discussed in the main paper, there exists a trade-off between the quality of generated motion and the capacity to retain the appearance of the initial sketch. To illustrate this trade-off, we conducted an experiment wherein we randomly selected three sketches from each class in our evaluation set (9 sketches in total). We then tested the impact of scaling the local learning rate within the range of 0.01 to 0.0001, keeping all parameters constant except for the local learning rate. Qualitative results are shown in the website, under the "Trade-off" section. Observe that as we move from the left (0.0001) to right (0.1), the motion in the animations increases, better aligning with the text prompt. However, this comes at the cost of preserving the original sketch's appearance. For example, observe how the fish and the crab undergo complete transformations when using a learning rate greater than 0.001. This trade-off introduces additional control for the user, who may prioritize stronger motion over sketch fidelity.

Furthermore, we assess the results using CLIP-based metrics (Fig. 13). As can be observed, increasing the learning rate leads to a smooth tradeoff between motion quality and sketch preservation. Working with learning rates $\in [0.001, 0.005]$ generally leads to a good compromise between the two aspects - though a user can choose a different working point according to their preferences.

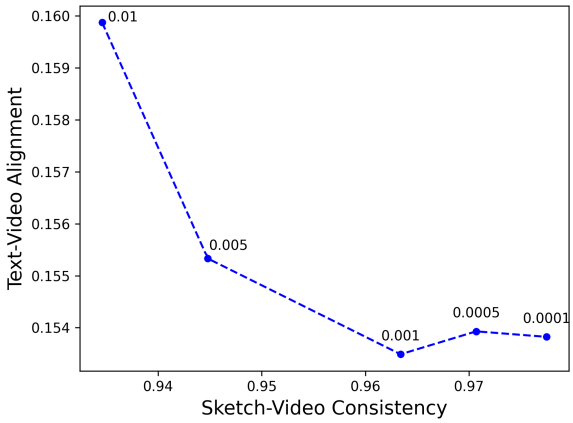


Figure 13. Investigation of the tradeoff between motion quality and sketch preservation. Increasing the local learning rates trades one aspect for another.

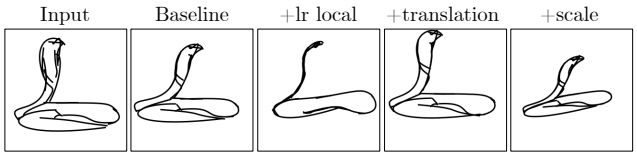


Figure 14. Hyperparameter effect. We show one representative frame from each video (the full videos and additional examples are provided in the website).

B.5. Hyperparameter effects

We demonstrate how changing different hyperparameters in our method can provide the user with additional control (see "Hyperparameter Effects" in the website). We observe different effects across various sketches, which may be attributed to the video model's prior or the initial sketch quality. Specifically, in the third column ("+lr local"), we showcase the impact of increasing the learning rate of the local path. As evident, in some cases (biking and butterfly), this improved the generated motion without significantly harming the sketch's appearance. However, in other cases (cobra and boat), increasing the local path's learning rate leads to a complete alteration of the original sketch. In the fourth and fifth columns we show the effect of increasing the translation and scale prediction weights. As observed, this indeed causes the objects to move more across the frame or change their scale.

B.6. Other text-to-video backbones

We investigate the performance of the model when we swap one text-to-video prior for another. In the main paper, we use ModelScope [82] as our text-to-video diffusion backbone. Here, we qualitatively evaluate the effect of replacing it with other text-to-video models. In particular, we look at a set of ZeroScope models, tuned across a range of resolutions and framerates. The results are shown in the supplementary videos (website section "Comparing Video Models"). Two representative examples are provided in Fig. 15. Our method generalizes to these models with no additional changes. However, note that different models do lead to different motion patterns, and some of them may result in different tradeoffs between the level of motion and the ability to preserve the sketch. For example, observe the cat (second row) which either wags their tail, raises its front legs, or does both, depending on the model. For some models (*e.g.* zeroscope v1-1 320s) the cat appears more deformed, and a user may prefer to use another working point on the local learning-rate axis in order to restore the shape.

C. Implementation and technical details

Here we outline additional details required to reproduce our work and experiments. We will release all code and image sets used for evaluations to facilitate further research and comparisons.

C.1. Sketch generation

Unless otherwise noted, all sketches presented in the main paper and the supplementary material were generated using CLIPasso [80]. CLIPasso is a method for automatically generating object sketches represented with cubic Bezier curves. In the majority of examples, we applied CLIPasso with the default settings, using 16 strokes. The sketch's canvas size is 256×256, and the strokes width is 1.5. It is important to note that our method can be employed with vector sketches created through alternative approaches, such as [22, 28, 39, 79], or even sketched by hand. For optimal performances, we recommend to represent the input sketch with cubic Bezier curves.

C.2. Additional training details

To improve stability in early training steps, we initialize $\mathcal{M}$ so that the predicted local displacements are small and the global transformations $\mathcal{T}^j$ are close to the identity matrix.

When sampling timesteps for the SDS loss, we follow DreamFusion [67] and avoid sampling very early or very late steps. In practice we sample the steps uniformly in the range $[50, 950]$.

When rendering the video frames for training we use a canvas size of 256×256, even when using text-to-video models trained with different aspect ratios. This limitation is primarily due to memory constraints. Lifting this restriction may aid in improving visual fidelity at the cost of higher VRAM requirements. We similarly restrict ourselves to 24 frames. Increasing this value can improve smoothness at the cost of additional memory. Our baseline method requires roughly 23GB of VRAM.

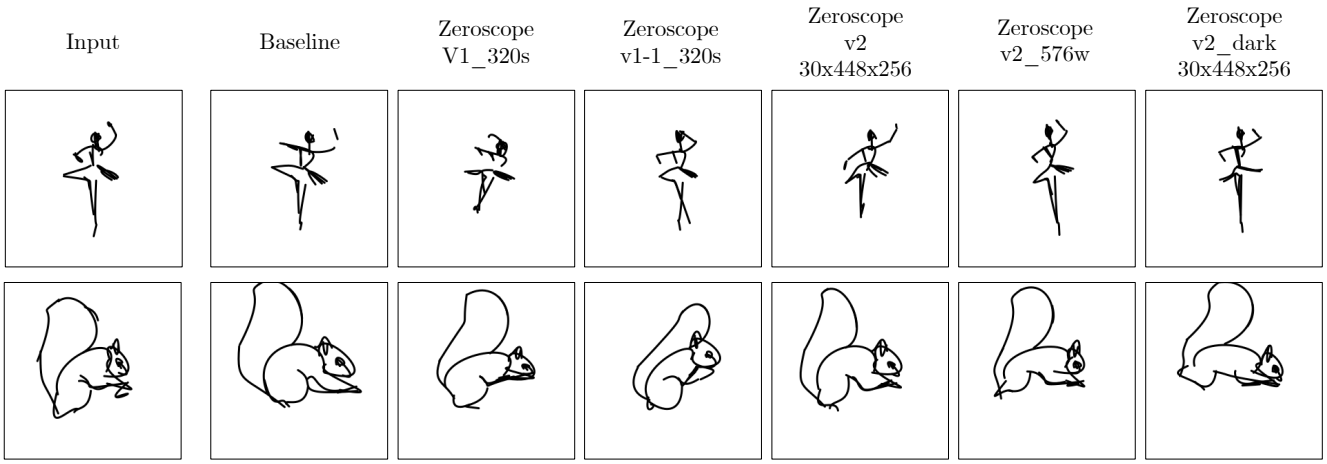


Figure 15. Other text-to-video backbones. We show the first frame from the results of five alternative text-to-video models. The full videos and additional examples are provided in the website. Observe that the choice of backbone model affects the output video in terms of both the sketch's appearance and the type of generated motion.

C.3. Evaluation details

C.3.1 Baseline implementations

When comparing to alternative methods, we used the following implementations:

- ModelScope: https://huggingface.co/spaces/damo-vilab/MS-Image2Video-demo/tree/main
- ZeroScope: https://huggingface.co/spaces/fffiloni/zeroscope-img-to-video/tree/main
- VideoCrafter: https://huggingface.co/spaces/VideoCrafter/VideoCrafter/tree/main
- Animated Drawings: https://sketch.metademolab.com/canvas
- Gen-2: https://research.runwayml.com/gen2

Note that Gen-2 is actively updated. We obtained our results on October 19th, 2023.

C.3.2 Evaluation metrics

For our sketch-to-video consistency metric we use OpenAI's CLIP ViT-B/32. For the text-to-video alignment metric we use Microsoft's xclip-large-patch14. This X-CLIP model expects 8 input frames, which are sampled uniformly from the generated video.

C.3.3 Evaluation data

In Tabs. 2 to 4 we provide the list of sketches used for our quantitative evaluations, along with their associated prompt.

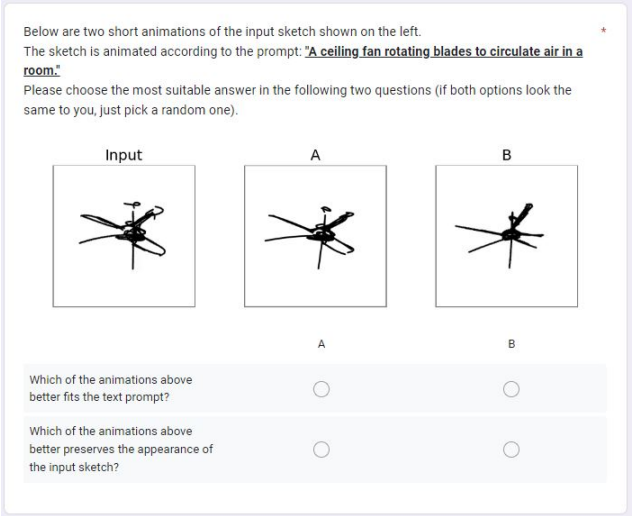


Figure 16. User study example question.

C.3.4 User Study

As discussed in section 5.2 of the main paper, we conduct a user study to validate our suggested components. The user study examines the sketch-to-video consistency and text-to-video alignment of the animations produced when disabling different components of our method. An example question is shown in Fig. 16. These questions were repeated for all the targets in the evaluation set, each time comparing our full method to a random choice of the ablation scenarios.

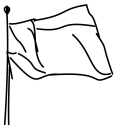
Table 2. Sketches, and prompts used for our quantitative evaluations for the ”animal” class.

	The penguin is shuffling along the ice terrain, taking deliberate and cautious step with its flippers outstretched to maintain balance.
	The goldenfish is gracefully moving through the water, its fins and tail fin gently propelling it forward with effortless agility.
	The crab scuttled sideways along the sandy beach, its pincers raised in a defensive stance.
	A galloping horse.
	The eagle soars majestically, with powerful wing beats and effortless glides, displaying precise control and keen vision as it maneuvers gracefully through the sky.
	A hummingbird hovers in mid-air and sucks nectar from a flower.
	A dolphin swimming and leaping out of the water.
	A butterfly fluttering its wings and flying gracefully.
	A gazelle galloping and jumping to escape predators.
	The squirrel uses its dexterous front paws to hold and manipulate nuts, displaying meticulous and deliberate motions while eating.

Table 3. Sketches, and prompts used for our quantitative evaluations for the ”human” class.

	The two dancers are passionately dancing the Cha-Cha, their bodies moving in sync with the infectious Latin rhythm.
	The boxer ducking and weaving to avoid his opponent’s punches, and to punch him back.
	The runner runs with rhythmic leg strides and synchronized arm swing propelling them forward while maintaining balance.
	The jazz saxophonist performs on stage with a rhythmic sway, his upper body sways subtly to the rhythm of the music.
	The ballerina is dancing.
	The biker is pedaling, each leg pumping up and down as the wheels of the bicycle spin rapidly, propelling them forward.
	A martial artist executing precise and controlled movements in different forms of martial arts.
	A surfer riding and maneuvering on waves on a surfboard.
	A figure skater gliding, spinning, and performing jumps on ice skates.
	A basketball player dribbling and passing while playing basketball.

Table 4. Sketches, and prompts used for our quantitative evaluations for the ”object” class.

	A waving flag fluttering and rippling in the wind.
	A parachute descending slowly and gracefully after being deployed.
	A wind-up toy car, moving forward or backward when wound up and released.
	A windmill spinning its blades in the wind to generate energy.
	A ceiling fan rotating blades to circulate air in a room.
	A clock hands ticking and rotating to indicate time on a clock face.
	The wine in the wine glass sways from side to side.
	The airplane moves swiftly and steadily through the air.
	The spaceship accelerates rapidly during takeoff, utilizing powerful rocket engines.
	The flower is moving and growing, swaying gently from side to side.