

# AniClipart: Clipart Animation with Text-to-Video Priors

RONGHUAN WU, City University of Hong Kong  
 WANCHAO SU, Monash University  
 KEDE MA, City University of Hong Kong  
 JING LIAO\*, City University of Hong Kong

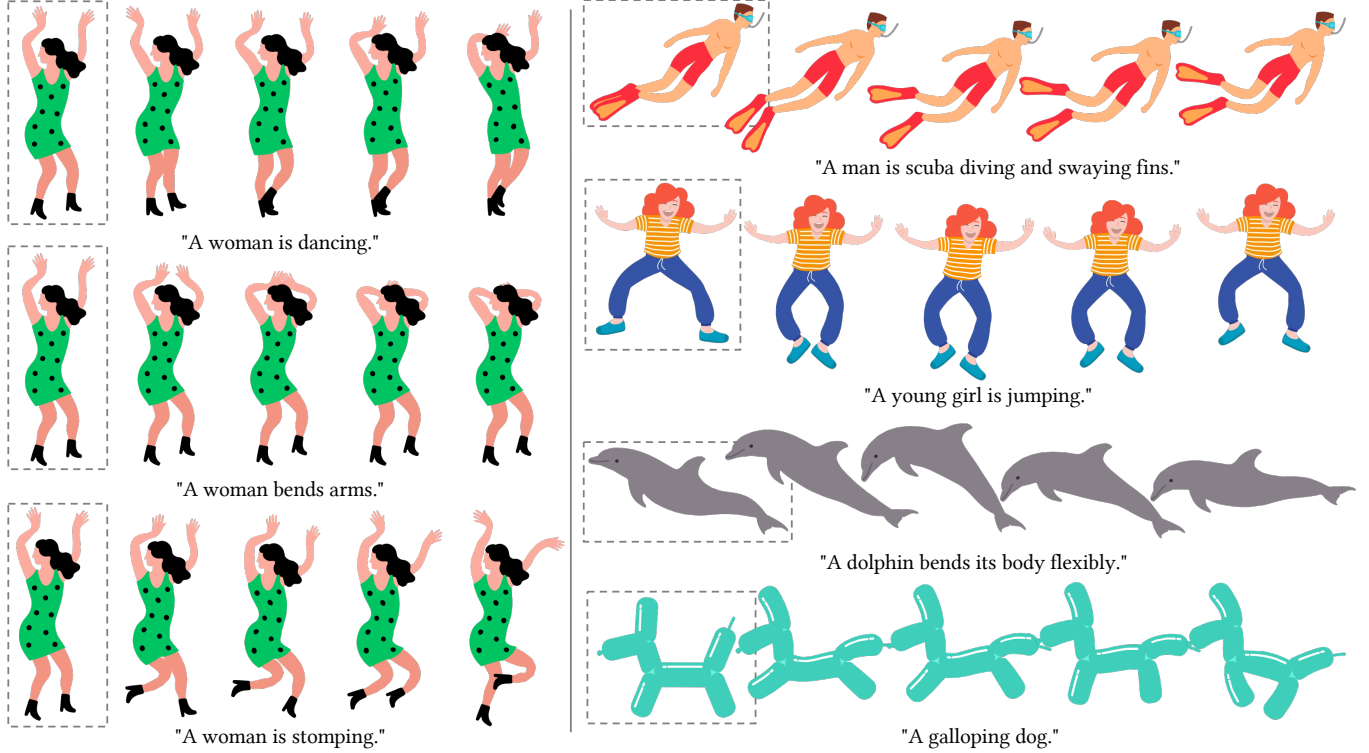


Fig. 1. Our AniClipart creates clipart animations guided by text prompts while preserving the visual identity of the input static clipart and achieving frame-to-frame consistency. The left panel displays different animations generated from the same clipart input, guided by different text prompts. On the right panel, we present animations with diverse input categories. We denote the initial clipart with a dashed-line box.

Clipart, a pre-made graphic art form, offers a convenient and efficient way of illustrating visual content. Traditional workflows to convert static clipart images into motion sequences are laborious and time-consuming, involving numerous intricate steps like rigging, key animation and in-betweening. Recent advancements in text-to-video generation hold great potential in resolving this problem. Nevertheless, direct application of text-to-video generation models often struggles to retain the visual identity of clipart images or generate cartoon-style motions, resulting in unsatisfactory animation outcomes. In this paper, we introduce AniClipart, a system that transforms static clipart images into high-quality motion sequences guided by text-to-video priors. To generate cartoon-style and smooth motion, we first define Bézier curves over keypoints of the clipart image as a form of motion regularization. We then align the motion trajectories of the keypoints with the provided

text prompt by optimizing the Video Score Distillation Sampling (VSDS) loss, which encodes adequate knowledge of natural motion within a pretrained text-to-video diffusion model. With a differentiable As-Rigid-As-Possible shape deformation algorithm, our method can be end-to-end optimized while maintaining deformation rigidity. Experimental results show that the proposed AniClipart consistently outperforms existing image-to-video generation models, in terms of text-video alignment, visual identity preservation, and motion consistency. Furthermore, we showcase the versatility of AniClipart by adapting it to generate a broader array of animation formats, such as layered animation, which allows topological changes. The project page is <https://aniclipart.github.io>.

Additional Key Words and Phrases: Clipart Animation, Text-to-Video Diffusion, Score Distillation Sampling, As-Rigid-As-Possible Shape Deformation.

\*Corresponding Author.  
 Authors' addresses: Ronghuan Wu, City University of Hong Kong, rh.wu@my.cityu.edu.hk; Wanchao Su, Monash University, wanchao.su@monash.edu; Kede Ma, City University of Hong Kong, kede.ma@cityu.edu.hk; Jing Liao\*, City University of Hong Kong, jingliao@cityu.edu.hk.

## 1 INTRODUCTION

Clipart, a collection of readily pre-made graphic elements, provides a convenient solution for swiftly enhancing visual content without

crafting custom artwork from scratch. Its user-friendliness and wide range of styles render it an ideal tool for enhancing the visual appeal of documents, presentations, websites, and other mediums. The animated clipart takes all these benefits and elevates them to the next level. Compared to the static counterpart, the animated clipart is particularly effective in grabbing users’ attention, making key messages easier to follow, and adding a spark of enjoyment to the content.

In traditional production, animating an existing clipart is delicate and laborious, including various steps like rigging, key-framing, in-betweening, and carefully considering timing and spacing. Recent advancements in Text-to-Image (T2V) synthesis (e.g., Stable Diffusion [Rombach et al. 2022]) have revolutionized the creation of high-quality clipart, shifting it from purely manual labor to a more automated process. However, automatically animating a given clipart is still underexplored.

The growing demand for animated cliparts and their labor-intensive creation process highlight the need for a system that can animate an existing clipart with minimal or no manual intervention. A natural solution is using recent Text-to-Video (T2V) models that accept a text prompt and an image simultaneously [Chen et al. 2023b, 2024; Xing et al. 2023b; Zhang et al. 2023]. However, these models prove inadequate for generating clipart animations for two reasons. First, there is a significant discrepancy in motion patterns between videos generated by existing large-scale T2V models and those in clipart animations. The training datasets for video generation models consist of natural-style videos that exhibit realism, complexity, and subtlety in motion. In contrast, cartoon-style clipart animations favor straightforwardness and simplicity, aiming for rapid conveyance of messages. Second, video generation models tend to synthesize videos that compromise the visual identity and distinct features of the original clipart, due to pixel-level distortions in the output videos, such as flickering backgrounds and blurred patches. Hence, the disparity in motion paradigms and pixel-level distortion render the state-of-the-art video generative models unsuitable for clipart.

In this paper, we present our system, *AniClipart*, which leverages pretrained large-scale T2V models to animate a given clipart and align it with a text prompt while addressing the aforementioned challenges. Inspired by the standard animation pipeline, we define keypoints on the clipart and distill the keypoints’ trajectory from the T2V model to generate animations. By driving the clipart using these keypoints, we successfully avoid pixel-level artifacts and better preserve the visual identity of the clipart. Moreover, by confining the motion to selected keypoints, we significantly reduce the complexity of motion and bridge the domain gap between different motion patterns.

Specifically, we assign Bézier curves as motion trajectories for each keypoint, determining the positions of keypoints in each keyframe and ensuring smooth transitions between frames. To align the motion with a text prompt, we propose the use of Video Score Distillation Sampling (VSDS) loss to optimize the parameters of the Bézier trajectories, distilling motion knowledge from a pretrained T2V diffusion model. Score Distillation Sampling (SDS), a technique introduced in Dreamfusion [Poole et al. 2022], extends the knowledge acquired from pretrained diffusion models to tasks beyond

their original scope. This capability is invaluable in fields with limited data availability, such as generating 3D models [Mildenhall et al. 2020] and vector graphics [Jain et al. 2022] from pretrained image diffusion models. To extend SDS from the image domain to the video domain, we input the animated clipart into a video diffusion model [Wang et al. 2023a] and assess the disparity between the current motion and the video diffusion model’s motion prior based on the text description. This enables us to achieve precise and contextually appropriate animation results.

Moreover, one key aspect of clipart animation is preserving the visual identity of the input. This requires retaining its local details and shape rigidity. To achieve this, in addition to the SDS loss, we propose incorporating a skeleton loss in the trajectory optimization process. This skeleton loss constrains the length variation of the skeleton formed by keypoints, thereby ensuring shape regularity. The optimization of both losses is integrated with a 2D shape deformation algorithm, specifically the As-Rigid-As-Possible (ARAP) shape manipulation [Igarashi et al. 2005], to preserve the rigidity of deformation. We further make the ARAP algorithm differentiable, allowing it to warp the clipart to a new pose defined by the updated keypoint positions, while also enabling backpropagation of gradients from the image-level loss to update the keypoints.

Extensive experiments and ablation studies demonstrate our AniClipart’s ability to generate vivid and attractive clipart animations from text descriptions across various subjects, including humans, animals, and objects. Our system also supports layered animation to handle motion patterns involving topological changes and self-occlusion. Our contributions in this work can be summarized as:

- (1) We introduce a novel system capable of generating animations for clipart based on textual descriptions, marking a step forward in automatic animation generation.
- (2) We successfully utilize VSDS loss to distill motion from large-scale T2V models and apply it to optimize keypoint trajectories, resulting in semantically meaningful motions while creating abstract cartoon-style clipart motion patterns.
- (3) We integrate shape deformation algorithms with our skeleton loss to effectively maintain the visual identity of the original clipart across the generated animation.

## 2 RELATED WORK

This section summarizes the prior works that closely relate to our approach. Given the breadth of topics related to 2D character animation, a comprehensive survey of each related topic is beyond the scope of this study. In this paper, we develop a system to animate existing 2D clipart (Section 2.1) based on text input. Instead of training a feed-forward network from scratch, we leverage a pretrained T2V model (Section 2.2) and employ score distillation sampling loss (Section 2.3) to extract motion.

### 2.1 Animating 2D Cartoon Characters

Figure 2 shows a simplified animation pipeline. We highlight three critical steps to animate 2D cartoon characters: rigging, key animation and in-betweening. Recent studies have concentrated on developing automated algorithms for these critical steps.

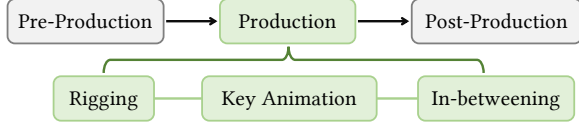


Fig. 2. A simplified animation pipeline.

*Rigging* is a time-consuming step in animation that constructs skeletons for characters. Along with skinning algorithms [Forstmann and Ohya 2006; Kavan et al. 2007; Le and Lewis 2019], transformations applied to the skeletons can be distributed across the entire shape. This rescues animators from having to draw every frame manually. Automatic rigging techniques can be classified into two categories. The first category maps a predetermined skeletal template onto the character [Baran and Popović 2007; Li et al. 2021a], but it lacks flexibility when characters are not compatible with the template. The second category can generate rigs for broader categories. Traditional approaches [Au et al. 2008; Huang et al. 2013; Tagliasacchi et al. 2012] analyze geometric features of the mesh to produce curve-skeletons but often ignore movable parts. In recent years, deep learning-based methods [Liu et al. 2019; Xu et al. 2020] demonstrate a robust capacity for rig generation, offering a viable alternative for animators.

*Key Animation*. To define character poses in keyframes, animators create a deformable puppet, typically represented by a triangular mesh, on the characters and manually adjust handles (e.g., skeletons and control points generated in rigging) across the keyframes. However, defining key poses is still challenging for those without animation skills [Fan et al. 2018]. Researchers have thus focused on developing methods to transfer motion to predefined puppets. Hornung et al. [2007] fitted a 3D model onto 2D characters and then animated them with 3D motion. Animated Drawings [Smith et al. 2023] creates rigged characters from children’s art, and maps predefined human motion for animation. Bregler et al. [2002] and DeJuan and Bodenheimer [2006] captured and transferred motion from existing cartoons, enriching the expressiveness of animations with diverse movements. Furthermore, recent advancements include extracting motion from videos. Live Sketch [Su et al. 2018] transfers video-derived motion trajectories to sketches, Pose2Pose [Willett et al. 2020] uses cluster algorithms to select key poses from video-tracked performances, and AnaMoDiff [Tanveer et al. 2024] warps the character according to the optical flow extracted from a driving video.

*In-betweening*. Inserting drawings between keyframes can transform choppy and disjointed animations into smooth and seamless sequences. A straightforward approach involves interpolating between keyframes with shape interpolation algorithms [Alexa et al. 2000; Baxter et al. 2009, 2008; Chen et al. 2013; Fukusato and Maejima 2022; Fukusato and Morishima 2016; Kaji et al. 2012; Whited et al. 2010]. Nevertheless, these algorithms often struggle to produce life-like movements [Fukusato et al. 2023]. On the other hand, recognizing animation in-betweening as a subset of frame interpolation opens the door to applying recent advancements in video frame interpolation techniques [Huang et al. 2022; Jiang et al. 2018; Liu et al. 2017; Lu et al. 2022; Niklaus and Liu 2018, 2020; Niklaus

et al. 2017a,b; Park et al. 2020; Reda et al. 2022; Sim et al. 2021; Xu et al. 2019]. However, Siyao et al. [2021] pointed out that animation videos possess their distinct features, such as flat color regions and exaggerated motions, which lead general video interpolation models to perform poorly. Consequently, there is a growing focus on developing algorithms tailored for the animation domain [Chen and Zwicker 2022; Li et al. 2021b; Siyao et al. 2023, 2021].

In our paper, we develop a system that automates these three crucial steps: we use a hybrid rigging algorithm to create accurate skeletons for both humans and objects. We select points along motion trajectories as the keyframes and refine them using video diffusion models. Additionally, the motion paths enable us to interpolate frames at any point to ensure smooth animation.

## 2.2 Text-to-Video Generation

Generating videos from text descriptions is challenging due to the scarcity of high-quality text-video pairs, the variable lengths of videos, and the high computational cost involved. As T2V generation extends the principles of T2I creation, the development of T2V models follows a trajectory similar to that of T2I: transformers [Hong et al. 2022; Kondratyuk et al. 2023; Villegas et al. 2022; Wu et al. 2022] and diffusion models [Gupta et al. 2023; Ho et al. 2022]. Regarding training strategy, one research direction involves training a T2V model from scratch utilizing both image and video data to learn their joint distribution [Ho et al. 2022; Singer et al. 2022]. Such a method demands significant training time and computational resources. Recent studies have explored the integration of additional temporal consistency modules into existing T2I models (e.g., Stable Diffusion [Rombach et al. 2022]), using text-video data pairs to train the newly-added layers (or finetune the entire model) [Blattmann et al. 2023; Ge et al. 2023; Girdhar et al. 2023; Guo et al. 2023; Singer et al. 2022; Yuan et al. 2024]. This training paradigm leverages the robust image prior of T2I models to maintain spatial consistency, while additional dedicated modules learn to extend temporal dimension, thereby accelerating convergence. However, attempting to animate clipart using these models often leads to pixel-level artifacts, such as distortion and blurriness, in the resulting videos. Moreover, since these models are primarily trained on natural videos, they are not well-suited for generating clipart animation. To address these issues, we propose a solution that involves distilling motion from T2V diffusion models and leveraging it to optimize the motion trajectories of keypoints defined on clipart. By adopting this approach, we can better regulate the motion and achieve the desired cartoon animation style required by clipart.

## 2.3 Score Distillation Sampling

T2I diffusion models are pretrained on an extensive collection of text-image pairs, providing a comprehensive understanding of relations between text descriptions and 2D images. Utilizing image diffusion models beyond image synthesis and transferring prior knowledge to new tasks, particularly those with limited annotated data, have become a prominent research area. DreamFusion [Poole et al. 2022] has shown the feasibility of generating 3D representations (e.g., NeRF [Mildenhall et al. 2020]) from a text description using only 2D image diffusion models, with a newly devised loss function

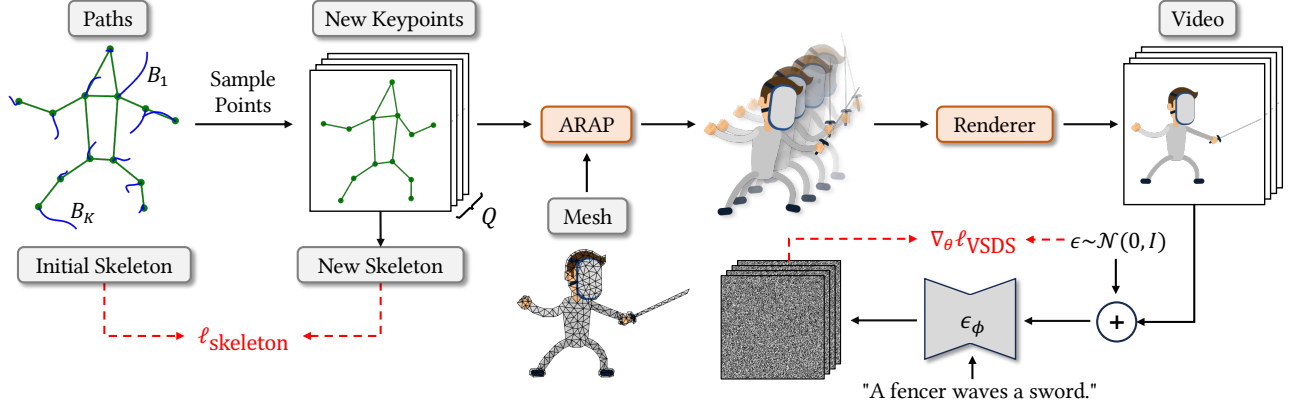


Fig. 3. The system diagram of AniClipart. There are  $K$  Bézier paths,  $\{B_i\}_{i=1}^K$ , serving as the motion trajectories of  $K$  keypoints. We select frame numbers  $Q$ , define timesteps, and sample points along the adjusted Bézier paths to determine new positions of keypoints. These new keypoints guide the differentiable ARAP algorithm to warp the triangular mesh, transforming the clipart to new poses. Subsequently, we use a differentiable renderer to convert the updated shapes into a video  $V$  and send it to a video diffusion model  $\epsilon_\phi$  to compute loss functions.

– score distillation sampling loss. SDS is similar to the diffusion model training loss but distinctively excludes the U-Net Jacobian term. This modification eliminates the need for computationally expensive backpropagation within the diffusion model and guides the optimization process toward aligning the images with the textual conditions. Subsequently, SDS loss is widely used to optimize other generation tasks, including artistic typography [Iluz et al. 2023; Tanveer et al. 2023], vector graphics [Jain et al. 2022], sketches [Qu et al. 2023; Xing et al. 2023a], meshes [Chen et al. 2023a] and texture maps [Metzer et al. 2023; Tsalicoglou et al. 2023]. The advent of T2V diffusion models [Bar-Tal et al. 2024; Chen et al. 2023b, 2024; Dai et al. 2023; Ni et al. 2023; Wang et al. 2023b] naturally projects the image SDS loss to extended dimensional applications. Recently, video SDS loss has been applied to create vector sketch animations [Gal et al. 2023]. This research involves predicting the positions of all points within a sketch for each frame and using VSDS loss to align the entire animation with a textual prompt. However, this technique falls short for clipart animation due to its inability to maintain shape rigidity and the resultant inconsistency in animation across frames. In our study, we apply VSDS loss specifically to refine the motion trajectories of keypoints.

### 3 ANICLIPART

In this section, we provide a detailed elaboration of our AniClipart. We begin with an overview of our method (Section 3.1), followed by the introduction of the implementation in three main aspects: clipart preprocessing (Section 3.2), Bézier-driven animation (Section 3.3) and loss functions (Section 3.4).

#### 3.1 Method Overview

In the clipart preprocessing stage, we detect keypoints, build skeletons, and construct a triangular mesh over the clipart for shape manipulation. In the Bézier-driven animation phase, we attach a Bézier curve to each of these keypoints, serving as the motion trajectories regulating these keypoints over time. When the keypoints

reach their new positions at a specific time, we use the differentiable as-rigid-as-possible shape deformation algorithm [Igarashi et al. 2005] to adjust the entire clipart to new poses according to the new keypoint positions. Such continuous deformation results in a series of consecutive frames, which can be naturally concatenated to create a video  $V$ . This video is then sent to a pretrained video diffusion model [Wang et al. 2023a] to compute the video score distillation sampling loss [Poole et al. 2022]. Along with a skeleton length loss to maintain the clipart’s shape integrity, we optimize the parameters of Bézier curves to ensure that the animation aligns with the provided text description. We show the overall workflow of our method in Figure 3 and explain the detail for each component in the following subsections.

#### 3.2 Clipart Preprocessing

Similar to the traditional animation procedure, the first step is character rigging, which detects keypoints over the clipart and builds skeletons between these points. State-of-the-art keypoint detection algorithms [Jiang et al. 2023; Mathis et al. 2018; Ng et al. 2022; Sun et al. 2023; Xu et al. 2022; Yang et al. 2023a; Ye et al. 2022] perform well in assigning predefined keypoints to articulated characters. Nonetheless, such methods are often limited to object categories within their training datasets. This constraint becomes evident when dealing with clipart, which encompasses various cartoon objects.

Therefore, we adopt a hybrid approach for keypoint detection. We use off-the-shelf UniPose [Yang et al. 2023b] to predict the keypoints and skeletons that have semantic meanings for articulated objects, such as humans (see Figure 4). UniPose unifies keypoint detection tasks into a comprehensive, end-to-end, prompt-driven framework capable of identifying keypoints across a diverse category of objects, including articulated, rigid, and soft objects. We choose this algorithm for its robust generalizability (e.g., its efficacy with cartoon-style images) and its fine-grained granularity, such as adding more details to a jaguar’s existing keypoints by giving a prompt like "tail". For broader categories, like sea animals and



Fig. 4. **Clipart preprocessing.** We use UniPose [Yang et al. 2023b] to detect keypoints and skeletons of articulated objects (e.g., humans).

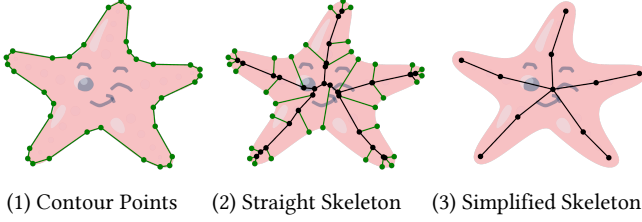


Fig. 5. Keypoint detection for broader categories.

plants, our keypoint detection algorithm involves three steps (see Figure 5 for illustrations):

- (1) Detecting contour points and connecting them to build boundary edges;
- (2) Generating straight skeletons by inwardly propagating edges in their perpendicular directions. For intersections where skeletons overlay are marked as keypoints;
- (3) Pruning unnecessary outer skeletons and merging adjacent intersections where the distance between them is shorter than a predetermined threshold. Commonly, we set the threshold as

$$0.7 \cdot \frac{1}{N_S} \sum_{i=1}^{N_S} \text{length}(s_i),$$

where  $s_i$  is the  $i$ -th skeleton and  $N_S$  is the total number of skeletons. Adjusting this threshold allows for the generation of skeletons with varying levels of complexity.

After keypoint detection, we use a triangulation algorithm [Shewchuk 1996] to create a mesh for deformation-based animation (see Figure 4).

### 3.3 Bézier-Driven Animation

Prior research [Gal et al. 2023] forecasts new positions for sketch control points at distinct timesteps. This approach, however, occasionally struggles to ensure temporal coherence throughout the animation frames. Additionally, as the complexity of the sketch escalates, the technique often falls short of preserving the fidelity of the original design. These inconsistencies produce significant distortion for clipart animation. Users expect animations to flow smoothly, and any shape distortion could jeopardize the overall quality of the animation.

To tackle the aforementioned problems, we define motion trajectories on a few keypoints rather than control all the points. We

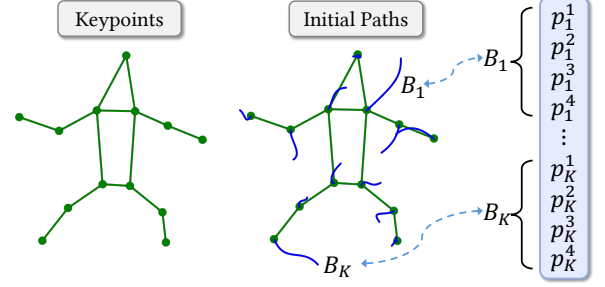


Fig. 6. **Motion path initialization.** We create a cubic Bézier curve for each keypoint, with the curve’s start point matching the initial keypoint position precisely. This means the animation begins with the object’s original position. The initial curves are short in our experiments, but we intentionally elongate them in this figure for clearer visualization.

employ 2D deformation algorithms to animate the entire clipart according to the keypoint motion. By this means, we reduce the search space for the motion creation while simultaneously maintain the clipart’s appearance. We particularly choose cubic Bézier curves as the motion trajectories due to their extensive application in the design field. Such parametric Bézier curves can be flexibly manipulated by altering the coordinates of control points. We assign a cubic Bézier curve for each keypoint, with each curve’s start point accurately aligned with the keypoint coordinate, ensuring the animation originates from the clipart’s initial pose, as shown Figure 6. For the remaining three points of each curve (i.e., two control points and an endpoint), we randomly initialize the positions, while making the initial curve length short, which ensures the initial animation presents moderate motions.

Formally, a clipart  $S$  contains  $K$  keypoints, denoted as  $\{k_i\}_{i=1}^K$ , corresponding to  $K$  cubic Bézier curves,  $\{B_i\}_{i=1}^K$ , where each curve  $B_i$  includes four control points,  $B_i = \{p_i^j\}_{j=1}^4$ . To monitor the progression of these keypoints over time, we define a sequence of timesteps,  $T = \{t^q\}_{q=1}^Q$ , with  $t^q \in [0, 1]$  and  $Q$  denotes the total number of frames in the animation. At every timestep  $t^q$ , we sample points on each cubic Bézier curve to determine the new positions of the keypoints, resulting in a set of updated positions,  $\{k_i^q\}_{i=1}^K$ . These new positions allow us to adjust the triangular mesh grid using the ARAP deformation algorithm [Igarashi et al. 2005]. ARAP allows users to deform the 2D shape by intuitively dragging several keypoints. It proposes a two-step closed-form algorithm that initially determines each triangle’s rotation and then adjusts the scale. ARAP has real-time performance while minimizing distortion across the triangular mesh. We further make the ARAP algorithm differentiable, enabling backpropagation of gradients. Subsequently, we warp the clipart, producing a new shape  $S^q$ . We then use a differentiable renderer  $\mathcal{R}$  to convert this altered shape into a pixel image, denoted as  $I^q = \mathcal{R}(S^q)$ . Repeating this procedure across all defined timesteps gives us a sequence of frames,  $I = \{I^q\}_{q=1}^Q$ . These frames are then temporally stacked to create a video  $V$  for further processing.

In our approach, we propose the arrangement of timesteps to create *looping animations*, drawing inspiration from the observation



that clipart animation frequently exhibits periodical movement patterns. These animations are designed to repeat continuously without any noticeable breaks at the end of the sequence. For a predetermined frame count  $Q$ , we begin by organizing timesteps in a monotonically increasing order but cap the sequence at  $Q/2$  frames. This yields  $\{t^q\}_{q=1}^{Q/2}$ , where  $t^1 = 0$ ,  $t^{Q/2} = 1$ , and  $t^1 < t^2 < \dots < t^{Q/2}$ . Afterward, to achieve a seamless looping effect, we invert this sequence and append it to the original, thereby generating a complete loop:

$$T_{\text{loop}} = \{t^0, \dots, t^{Q/2}, t^{Q/2}, \dots, t^0\}. \quad (1)$$

For our clipart examples,  $T_{\text{loop}}$  successfully creates animations with natural and rhythmic motions. For animations that do not require looping, designers are free to choose  $T_{\text{no-loop}} = \{t^q\}_{q=1}^Q$ , where  $t^1 = 0$ ,  $t^Q = 1$ , and  $t^1 < t^2 < \dots < t^Q$ , allowing for flexibility in animation styles.

### 3.4 Loss Functions

We use video SDS loss to animate a clipart with user-provided text input. To ensure the structural integrity of the original clipart, we also incorporate a skeleton preservation component in the total loss function.

**Video SDS Loss.** Unlike the image SDS loss, which focuses on augmenting a 2D image with additional content based on a single image input, the VSDS loss takes a video as input. It suggests how to add content across a sequence of images while simultaneously maintaining consistency across frames. In our setting, the updated motion trajectories are used to create a video  $V = \{I^q\}_{q=1}^Q \in \mathbb{R}^{h \times w \times Q}$ . We select an intermediate diffusion timestep  $t$  and infuse the video  $V$  with a randomly sampled noise  $\epsilon$ , which results in  $V_t = \alpha_t V + \sigma_t \epsilon$ , where  $\alpha_t$  and  $\sigma_t$  are parameters from the noising schedule. Following this, we pass the noise-injected video  $V_t$  and the text condition  $c$  to a pretrained T2V diffusion model [Wang et al. 2023a]. The model will output a predicted noise  $\epsilon_\phi(V_t, t, c)$ . Then, we compute the VSDS loss as follows:

$$\nabla_\theta \ell_{\text{VSDS}} = \mathbb{E}_{t, \epsilon} \left[ w(t) (\epsilon_\phi(V_t, t, c) - \epsilon) \frac{\partial V}{\partial \theta} \right], \quad (2)$$

where  $w(t)$  is a constant that relies on  $\alpha_t$  and  $\theta$  is the control points of cubic Bézier curves. This loss measures the discrepancy between the model's predicted noise and the actual noise  $\epsilon$ , steering the Bézier-driven animation towards alignment with the text description. The optimization process updates the shape of Bézier curves gradually, and is conducted iteratively until it reaches convergence.

**Skeleton Loss.** Optimizing motion trajectories using  $\nabla_\theta \ell_{\text{VSDS}}$  produces animations that align with the text input. Nevertheless, the solely VSDS-incorporated optimization process encounters difficulties in preserving the clipart's authentic appearance in the animation. To preserve the structure of the objects, we leverage the skeleton extracted by the keypoint detection algorithms (Section 3.2) to guarantee minimal deviation in skeletal component lengths from their original measurements. Suppose there are  $N_S$  skeletal components,  $\{s_i\}_{i=1}^{N_S}$ , in the original structure, we can get a new

skeleton,  $\{s_i^q\}_{i=1}^{N_S}$ , at a specific frame  $q$  based on the updated keypoint positions  $\{k_i^q\}_{i=1}^K$ . The overall skeleton loss is:

$$\ell_{\text{skeleton}} = \frac{1}{N_S} \sum_{q=1}^Q \sum_{i=1}^{N_S} (\text{length}(s_i^q) - \text{length}(s_i))^2. \quad (3)$$

**Total Loss and Optimization.** The weighted average of two terms then defines the final loss function:

$$\ell_{\text{total}} = \nabla_\theta \ell_{\text{VSDS}} + \lambda \ell_{\text{skeleton}}, \quad (4)$$

where  $\lambda = 25$  serves as a weighting factor to balance the relative significance of text alignment and the preservation of appearance. Given that the ARAP [Igarashi et al. 2005] is a closed-form algorithm that solves equations to determine new positions of vertices on triangular meshes, our entire process is differentiable, and the gradient can be backpropagated to update the motion trajectories.

## 4 EXPERIMENTS

To evaluate the effectiveness of our proposed *AniClipart*, we conducted extensive quantitative and qualitative experiments to assess our method's superiority over the alternative solutions. We organize this section as follows: We first introduce the experimental setup of our system (Section 4.1). After introducing the evaluation metrics (Section 4.2), we present the comparison between our model and a closely related study [Gal et al. 2023], as well as leading T2V generation models, highlighting the distinctions and contributions of our approach within the clipart field (Section 4.3). Furthermore, we present the outcomes of ablation studies to validate our method's superiority over the substitutional techniques in the animating procedure (Section 4.4). We also show our method's capability in handling more challenging topology-changing cases by incorporating layer information (Section 4.5).

### 4.1 Experiment Setup

We collected 30 cliparts from Freepik\*, including 10 humans, 10 animals, and 10 objects, with each clipart resized into  $256 \times 256$ . We optimized the positions of control points for Bézier curves with an Adam [Kingma and Ba 2014] optimizer for 500 steps, with a learning rate of 0.5. We used ModelScope text-to-video [Wang et al. 2023a] as the diffusion backbone, setting the entire animation to 24 frames and adjusting the SDS guidance scale to 50. The entire process of one clipart animation, conducted on an NVIDIA RTX A6000, requires about 25 minutes with 26GB memory consumption.

Typically, clipart is stored and distributed in two formats: bitmap and vector graphics. In our experiments, we conducted evaluations using clipart in the Scalable Vector Graphics (SVG) format. Our decision to focus on vector representation yields enhanced visual perception quality and more diverse outcomes: the scalability of SVG facilitates the production of high-resolution frames, while the inherent layered structure enables us to create complex animations with topological changes. Each SVG file includes multiple geometric shapes known as *Paths*, which are controlled parametrically by sequences of 2D control points. In our animation process, we update the control points according to the barycentric coordinates after adjusting the triangular shape with the ARAP algorithm. Then we

\*<https://www.freepik.com/>

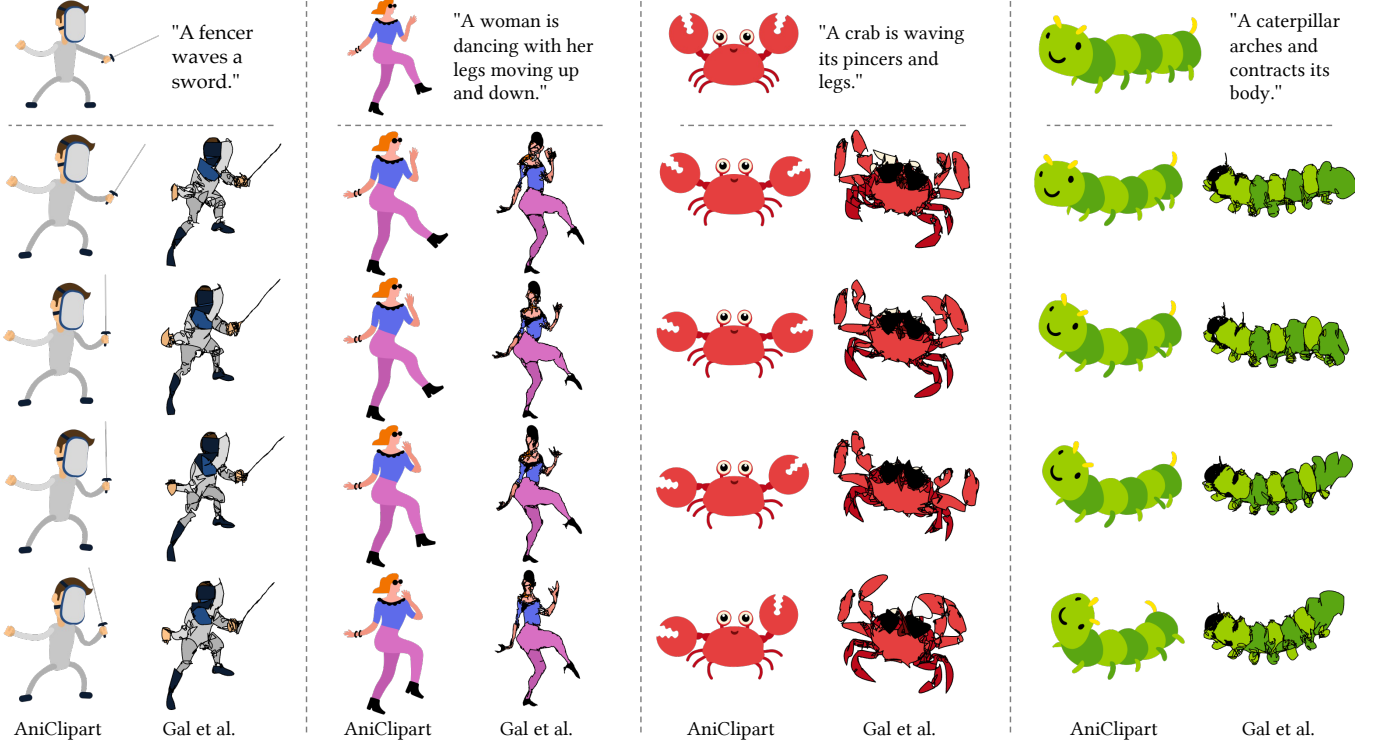


Fig. 7. **AniClipart versus Gal et al. [2023]**. We select four consecutive frames from the output animation to illustrate the differences. Our method effectively maintains the original shape, producing reasonable and visually pleasing animations. On the other hand, the approach by Gal et al. drastically alters the characters’ appearance and lacks continuity and consistency across frames.

use DiffVG [Li et al. 2020] to convert clipart from vector graphics format to bitmap, which is subject to video diffusion models. Our method is not limited to vector clipart; rasterized clipart can be processed using the same procedure with comparable output quality. The primary distinction lies in the warping operation for bitmap images, wherein pixel patches (instead of control points) are relocated to new positions.

## 4.2 Evaluation Metrics

To evaluate the quality of AniClipart, we conducted assessments at both the bitmap-level and vector-level of the generated animations.

**Bitmap-level Metrics.** To compare our system with state-of-the-art methods, we focus on two objectives for evaluating the quality of the generated animation: (1) maintaining the visual identity of the input clipart and (2) ensuring alignment between the text descriptions and the resulting animations. For visual identity preservation, we employed CLIP [Radford et al. 2021] to calculate the average cosine similarity scores between the feature representations of the input clipart image and each frame within the generated animation. To measure the alignment between the videos and text descriptions, we utilized X-CLIP [Ni et al. 2022], an extension of the CLIP model adapted for the video domain, to compute similarity scores.

**Vector-level Metrics.** Consistent with prior research [Gal et al. 2023], we observed that CLIP and X-CLIP fail to capture the nuanced

differences between animations produced in the ablation study. Consequently, we leveraged vector graphics outputs as a more precise means of evaluation. To be more specific, we computed three distinct metrics:

- (1) *Dynamism*. This refers to the average trajectory length of all keypoints in an animation, as a measure of the motion magnitude in the results. Notably, when the keypoints’ locations are predicted directly for each frame (see "w/o Trajectory" in Figure 9), we connect two keypoints in consecutive frames as the pseudo-trajectory.
- (2) *Temporal Consistency*. A vector graphic’s shape is defined by a set of control points along the contours. We calculate the Hausdorff distance (represented by  $D_H$ ) between two sets of control points in adjacent frames. We then use  $1 - D_H$  as the similarity score. We average these scores throughout the animation sequence to determine the level of consistency.
- (3) *Distortion*. We first calculate the curvature of each point in the initial clipart. For each frame, we recalculate the curvature, and the difference between the initial curvature and the curvature for each frame determines the level of distortion.

## 4.3 Comparison to State-of-the-Art Methods

Given the scarcity of existing works on direct clipart animation, we compared our system to two alternatives: (1) a prior work [Gal et al.

Method	Identity Preservation $\uparrow$	Text-Video Alignment $\uparrow$
Ours	<b>0.9414</b>	<b>0.2071</b>
Gal et al.	0.8379	0.1879
ModelScope	0.8618	0.2021
VideoCrafter2	0.8403	0.1992
DynamiCrafter	0.8008	0.1741
I2VGen-XL	0.8813	0.1999

Table 1. Quantitative comparisons to Gal et al. [2023] and state-of-the-art T2V models.

2023] that concentrates on animating vector sketches, and (2) the state-of-the-art T2V generative models.

**AniClipart versus Gal et al. [2023].** Both our AniClipart and Gal et al.’s method aim to extract motion priors using pretrained T2V diffusion models with VSDE losses. Despite this commonality, there are two major differences. A key difference is the absence of a rigidity constraint in Gal et al.’s method, which leads to noticeable alterations in the character’s appearance over different frames, as shown in Figure 7. While minor distortions are acceptable in abstract sketches, they become catastrophic for clipart with delicate parametric edges when animating. Gal et al.’s algorithm drastically modifies the appearance of all characters, rendering it unsuitable for clipart applications. Our method incorporates the ARAP deformation algorithm to explicitly uphold the rigidity of the clipart, thereby maintaining a consistent appearance through all frames and achieving the highest CLIP similarity scores, as shown in Table 1.

Another key distinction lies in the optimization target. Gal et al.’s method predicts new positions for *all* control points in a sketch, whereas ours focuses on altering only a small number of detected keypoints. Our method refines a motion trajectory for each keypoint, ensuring smooth motion regardless of the complexity or the quantity of control points in the original clipart. The number of keypoints is orders of magnitude less than that of the control points in the entire sketch (ours 13 versus Gal et al.’s 1067 for the fencer example), effectively reducing the searching space of the optimization process. Our keypoint-focused approach with the ARAP effectively maintains the integrity of the *in-frame* appearance and produces semantically consistent motion across the whole clipart. Additionally, predicting new positions for control points (or keypoints) can sometimes lead to inconsistent movements across consecutive time steps (see the supplementary video). In contrast, our method of distributing timesteps along a Bézier curve (Section 3.3) successfully eliminates the above issue, ensuring an *inter-frame* motion consistency.

**AniClipart versus T2V Models.** In our experiments, we rasterized vector clipart into bitmap images and incorporated ModelScope [Wang et al. 2023a], VideoCrafter2 [Chen et al. 2024], DynamiCrafter [Xing et al. 2023b] and I2VGen-XL [Zhang et al. 2023] to generate respective videos. In particular, ModelScope, the video backbone model of our method, cannot accept text descriptions and images simultaneously. Therefore, we mixed the initial image with random noise and sent the noise-injected image to ModelScope to create videos. We set the blending ratio at 0.84. A higher ratio can

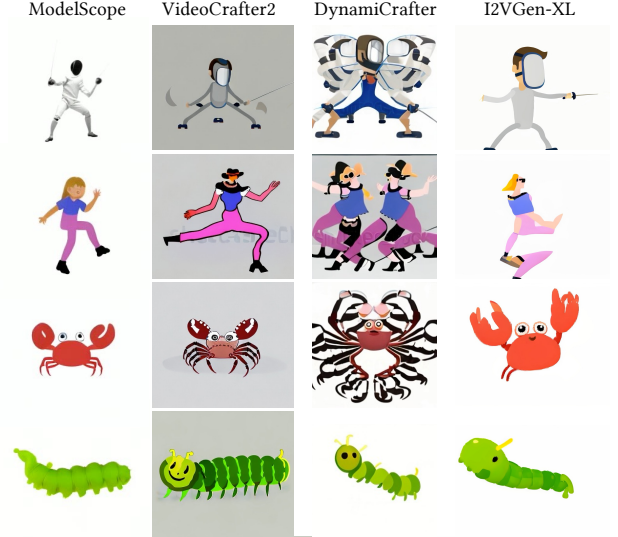


Fig. 8. **Results of T2V models.** These methods only preserve the semantics of the input clipart but fail to preserve the details, leading to low identity preservation scores. Furthermore, these approaches frequently yield animations with limited movement, resulting in inferior text-video alignment.

create larger motions but at the expense of losing visual identity, while a lower ratio can preserve the visual identity better but create almost static objects. We chose the latter three models because of their state-of-the-art video generation performances and ability to simultaneously accept text and image as inputs. In Figure 8, we show the representative frames in the videos created by T2V models. We observe two primary issues: (1) T2V models often generate some pixel-level artifacts, including shape distortion and blurred patches, thus cannot well preserve the original character’s visual identity, leading to lower identity preservation scores compared to ours. (2) The motion generated by T2V models is less pronounced compared to that of our system. Specifically, ModelScope strikes a decent balance between identity preservation and motion generation with a carefully chosen blending ratio but sometimes fails to drive the character to move at all, resulting in an inferior score in text-video alignment (Table 1). VideoCrafter2 may produce reasonable videos for a few cliparts. But in many cases, it fails to add motion to the objects. DynamiCrafter tends to add extra shapes and textures to the input clipart without introducing noticeable motions, thereby having the lowest text-video alignment score in Table 1. In addition, I2VGen-XL can produce moderate animations (e.g., the fencer) but at times drastically changes the appearance of the subjects (e.g., the dancing woman). Consequently, the text-video alignment scores of these models do not match up to the standards set by our methods.

**User Study.** We conducted a subjective user study to evaluate the perceptual quality of our AniClipart over other comparing methods. We excluded the Gal et al. [2023] in this study because their algorithm’s GPU memory usage increases in proportion to the number of control points. This is a problem because vector clipart often has more than 1000 points, which requires a significant amount of GPU resources. The user study comprised two distinct tasks to visually



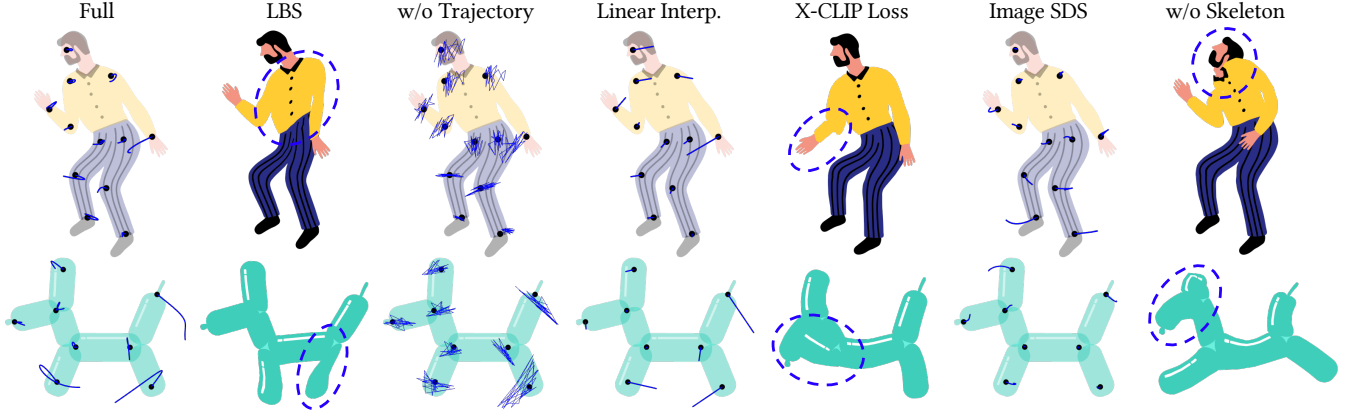


Fig. 9. **Results of ablation study.** Eliminating key components from our system could lead to animations with restricted movements (e.g., Linear Interp. and Image SDS), shape distortions (e.g., LBS, X-CLIP Loss and w/o Skeleton), and inconsistencies across frames (e.g., w/o Trajectory). Motion details are rendered by the overlaid blue trajectories, and shape details are highlighted by dash-lined circles.

evaluate (1) the preservation of visual identity and (2) the alignment between text and video. In the first task, we presented 30 static cliparts to participants, with each clipart followed by five animations created by AniClipart, ModelScope, VideoCrafter2, DynamiCrafter and I2VGen-XL, respectively. Each participant’s task was to identify the animation that best preserved the visual identity of the original clipart. The second task presented the participants with 30 textual prompts, each followed by the same set of five animations as in the first task. Here, we asked participants to select the animation they believed most closely matched the text prompt.

We conducted the user study via an online questionnaire, recruiting 48 participants in the study. To analyze the results, we calculated the average selection ratio of users across the 30 questions for both tasks, with the results presented in Table 2. The results clearly show that AniClipart significantly outperforms four other methods in terms of visual identity preservation due to the deformation algorithm in our system. Regarding text-video alignment, ModelScope occasionally produces reasonable animations that maintain identity, benefiting from a carefully optimized blending ratio. However, the other three T2V models often fail to produce noticeable motions that align with the text, revealing a limitation in their ability to ensure text-video coherence.

	User Selection% $\uparrow$	
	Identity Preservation	Text-Video Alignment
Ours	93.61	82.57
ModelScope	2.15	10.35
VideoCrafter2	0.97	2.08
DynamiCrafter	1.39	1.81
I2VGen-XL	1.88	3.19

Table 2. Subjective user study results.

#### 4.4 Ablation Study

In this section, we elaborate on the ablation study conducted to validate the effect of each essential component of our system, with the quantitative results detailed in Table 3. We also show a qualitative comparison of the ablation study in Figure 9. In the following of this subsection, we describe each component in detail.

**ARAP Deformation.** To analyze the effectiveness of ARAP incorporation, we substituted the deformation module with Linear Blend Skinning (LBS), which stands as another prevalent algorithm in shape deformation, where the new position of each point is determined by the spatially weighted transformations of the keypoints. In our study, we implemented LBS with bounded biharmonic weights [Jacobson et al. 2011]. Nonetheless, LBS frequently produces animations with unrealistic deformations, such as the shoulder squeezing in the dancing man case and leg distortion in the dog case in Figure 9. In Table 3, LBS also has higher distortion score when compared to ours. For settings without any 2D deformation algorithm employed, the generation aligns with the settings of the comparison between ours and Gal et al. [2023] (see Figure 7), which is inferior to our method.

**Bézier-Driven Animation.** We define Bézier curves as the motion trajectories for keypoints, aiming to achieve smooth and fluid animations. To understand their significance, we eliminated all motion paths and predicted the keypoints’ new positions in each frame. This setting could create animations matching the text description, but unfortunately, it failed to preserve frame-to-frame consistency, resulting in noticeable flickering effects ("w/o Trajectory" entries and Figure 9). Its dynamism score is the highest in Table 3 since the positions of keypoints change drastically in each frame, leading to the lowest frame consistency score. We further linked the keypoints in two frames to visualize the pseudo-trajectories in Figure 9, validating the significant inconsistency across frames. When we replaced Bézier curves with straight lines, the generated motions were overly simple. We refer to this setting as "Linear Interp." in the quantitative and qualitative results. Notably, such straight-line guided results presented the highest consistency score and the lowest distortion

	Dyna. $\uparrow$	Cons. $\uparrow$	Dist. $\downarrow$
Full	20.87	0.9673	50.98
LBS	17.13	0.9785	60.56
w/o Trajectory	<b>102.66</b>	0.9452	65.22
Linear Interp.	8.96	<b>0.9814</b>	<b>42.42</b>
X-CLIP Loss	8.16	0.9760	108.81
Image SDS	7.57	0.9779	59.07
w/o Skeleton	16.18	0.9675	265.92

Table 3. Quantitative results of the ablation study with "Dyna.", "Cons." and "Dist." representing Dynamism, Temporal Consistency, and Distortion, respectively. Our proposed method achieves a well-balanced trade-off among these three metrics.

rate in Table 3; we attribute such phenomenon to their limited motions, with little shape changes in the results. For instance, Figure 11 displays the optimized trajectories with only two control points (*i.e.*, linear interpolation), where the motion shows limited range and the appearance shows little change. Please find the side-by-side comparison in the accompanying video to better compare the motions produced in the animation results. These two alternative comparison methods clearly show that cubic Bézier curves are important in creating realistic and lifelike movements.

**Loss Functions.** The video SDS loss  $\nabla_{\theta} \ell_{\text{VSDS}}$  is crucial for generating motion trajectories that are in harmony with motion priors from text-to-video diffusion models. Replacing VSDS with another temporal-aware loss function, such as X-CLIP loss [Ni et al. 2022], does not produce reasonable motions aligned with the text prompt. Instead, the resulting animation displays unusual movements and significant distortions. Additionally, switching to an image-based SDS loss  $\nabla_{\theta} \ell_{\text{SDS}}$  compromised the animation’s relevance to the text input (the lowest dynamism score in Table 3). Furthermore, omitting skeleton loss  $\ell_{\text{skeleton}}$  results in unnatural limb proportions, diverging from human anatomical accuracy, see the visual details in Figure 9 and the highest distortion in Table 3.

#### 4.5 Extensions

We propose two extended implementations to our system to further increase the variety and quality of animations produced.

**Layered Animation.** We create a triangular mesh for clipart deformation in the clipart preprocessing stage (Section 3.2). However, such a single mesh may struggle to adequately handle cases involving self-occlusion and topological changes in animation. To enhance the capability of our system in solving the above-mentioned problem, we suggest shifting towards multi-layer animations through an enhanced clipart preprocessing strategy, layer separation, which involves:

- (1) partitioning the vector clipart into distinct groups;
- (2) independently identifying keypoints and skeletons within each group;
- (3) establishes joint points in different layers’ overlapping regions to prevent detachment.

We show an example in Figure 10 to illustrate the improvement of the clipart layers and the grouping operation.

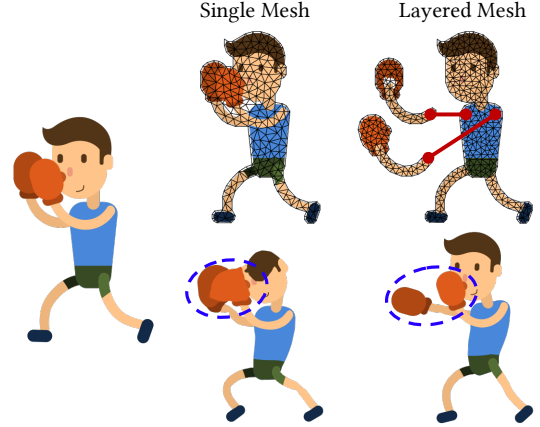


Fig. 10. **Layered animation.** The separation of different layers enables the animation involving topology changes. The left side shows the single-layered mesh and its corresponding animation result, presenting a distorted appearance. The right side illustrates the layered mesh anchored by two keypoints, producing animations with more natural motion.

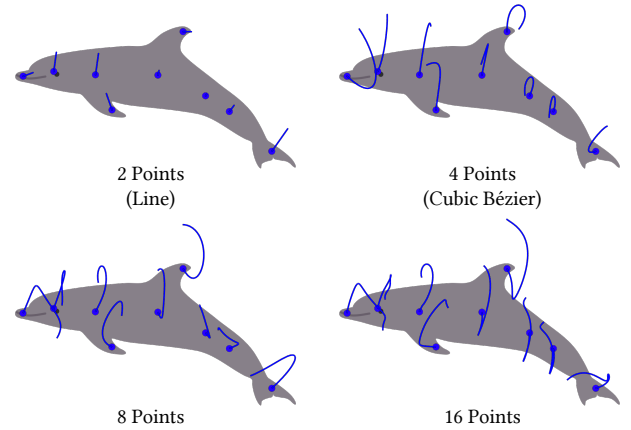


Fig. 11. The animation of optimized Bézier trajectories can be made smoother by increasing the number of control points.

**Higher-Order Bézier Trajectory.** In our experiments, we have demonstrated that cubic (*i.e.*, controlled by four points) Bézier curves effectively guide the motion trajectories of keypoints to produce plausible animations. It is worth noting that the motion trajectories can be defined by higher-order Bézier curves. Introducing additional control points enables the optimization to create more detailed motions. We include such examples for comparison in Figure 11, showcasing optimized trajectories with different numbers of control points. After comparing the generated animations (see the supplementary video), we conclude that cubic Bézier curves are adequate for creating both practical and visually pleasing animations. Although higher-order Bézier curves can incorporate finer details into animations, they involve more Bézier controlling parameters and necessitate a larger number of frames (*e.g.*, 36 or 48) in optimization, leading to prolonged time and higher computation resources.

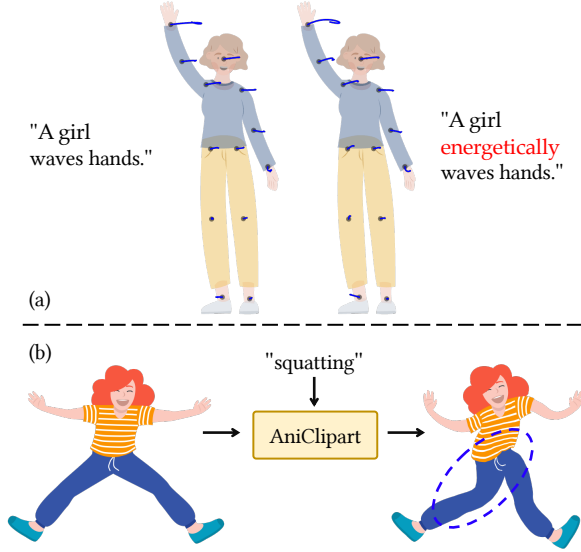


Fig. 12. **Limitations.** (a) Constrained by the capability of the T2V model, our AniClipart outputs similar motion trajectories, ignoring the variability in the provided text prompts. (b) The limited number of frames in the T2V model makes it challenging to animate a jumping girl into a squatting position.

in creating animations. We leave such an "expressiveness versus complexity" trade-off to designers in actual scenarios.

## 5 CONCLUSION AND DISCUSSION

In this paper, we propose a system that aims to animate a given clipart based on a provided text description. Our approach involves defining keypoints on the clipart and utilizing Bézier curves as motion trajectories to govern the movement of these keypoints, resulting in seamless and smooth motions. An important aspect of our method is that it generates motion trajectories without requiring additional training data. Instead, we extract motion priors from a pretrained text-to-video diffusion model using VSIDS loss. To animate the clipart while preserving its visual identity, keypoint trajectories are also constrained by a newly proposed skeleton loss and then used to drive the clipart using ARAP deformation. Extensive experiments demonstrate the system’s capability to produce high-quality clipart animations.

Despite the visually pleasant results produced, our AniClipart suffers from several limitations. Our method is constrained by the video backbone models. For instance, these models lack the ability to produce videos that precisely align with detailed textual descriptions. As illustrated in Figure 12(a), we present optimized trajectories based on two different prompts. Nonetheless, the term "energetically" fails to be accurately reflected in the trajectories’ lengths, resulting in two motions of similar dynamism. Fortunately, the VSIDS loss is not dependent on the underlying model, indicating a more sophisticated T2V model would help alleviate this problem.

Our system may face challenges when generating motions that significantly deviate from the original clipart pose, see Figure 12(b). The VSIDS loss in such cases has two tasks: deforming the shapes

to align with the text prompts and driving motion. However, the current video models do not have enough frames to achieve this transition. To overcome this challenge, users may manually drag the keypoints and deform the shape to align with the target text before starting the optimization process.

For future work, we have plans to explore 3D extensions. Our framework for optimizing motion trajectories by distilling T2V models can be directly integrated into 3D models using suitable shape deformation and differentiable rendering algorithms. This extension will enable us to animate 3D characters and objects.

## REFERENCES

- Marc Alexa, Daniel Cohen-Or, and David Levin. 2000. As-rigid-as-possible shape interpolation. *Conference on Computer Graphics and Interactive Techniques*, 1–8.
- Oscar Kin-Chung Au, ChiewLan Tai, HungKuo Chu, Daniel Cohen-Or, and TongYee Lee. 2008. Skeleton extraction by mesh contraction. *ACM Transactions on Graphics* 27, 3, 1–10.
- Omer Bar-Tal, Hila Chefer, Omer Tov, Charles Herrmann, Roni Paiss, Shiran Zada, Ariel Ephrat, Junhwa Hur, Yuanzhen Li, Tomer Michaeli, Oliver Wang, Deqing Sun, Tali Dekel, and Inbar Mosseri. 2024. Lumiere: A Space-Time Diffusion Model for Video Generation. *arXiv preprint arXiv:2401.12945*.
- Ilya Baran and Jovan Popović. 2007. Automatic rigging and animation of 3d characters. *ACM Transactions on Graphics* 26, 3, 72–es.
- William Baxter, Pascal Barla, and Ken Anjyo. 2009. N-way morphing for 2D animation. *Computer Animation and Virtual Worlds* 20, 2-3, 79–87.
- William Baxter, Pascal Barla, and Ken-ichi Anjyo. 2008. Rigid shape interpolation using normal equations. *International Symposium on Non-Photorealistic Animation and Rendering*, 59–64.
- Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, SeungWook Kim, Sanja Fidler, and Karsten Kreis. 2023. Align Your Latents: High-Resolution video synthesis with latent diffusion models. *IEEE Conference on Computer Vision and Pattern Recognition*, 22563–22575.
- Christoph Bregler, Lorie Loeb, Erika Chuang, and Hrishi Deshpande. 2002. Turning to the Masters: Motion capturing cartoons. *ACM Transactions on Graphics* 21, 3, 1–9.
- 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. 2023b. Videocrafter1: Open diffusion models for high-quality video generation. *arXiv preprint arXiv:2310.19512*.
- Haoxin Chen, Yong Zhang, Xiaodong Cun, Menghan Xia, Xintao Wang, Chao Weng, and Ying Shan. 2024. VideoCrafter2: Overcoming Data Limitations for High-Quality Video Diffusion Models. *arXiv preprint arXiv:2401.09047*.
- Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. 2023a. Fantasia3d: Disentangling Geometry and Appearance for High-Quality Text-to-3D Content Creation. *arXiv preprint arXiv:2303.13873*.
- Renjie Chen, Ofir Weber, Daniel Keren, and Mirela Ben-Chen. 2013. Planar shape interpolation with bounded distortion. *ACM Transactions on Graphics* 32, 4, 1–12.
- Shuhong Chen and Matthias Zwicker. 2022. Improving the perceptual quality of 2d animation interpolation. *European Conference on Computer Vision*, 271–287.
- Zuozhuo Dai, Zhenghao Zhang, Yao Yao, Bingxue Qiu, Siyu Zhu, Long Qin, and Weizhi Wang. 2023. Fine-Grained Open Domain Image Animation with Motion Guidance. *arXiv preprint arXiv:2311.12886*.
- Christina N. DeJuan and Bobby Bodenheimer. 2006. Re-using Traditional Animation: Methods for semi-automatic segmentation and inbetweening. *Eurographics symposium on Computer animation*, 223–232.
- Xinyi Fan, Amit H. Bermanto, Vladimir G. Kim, Jovan Popović, and Szymon Rusinkiewicz. 2018. Tooncap: A layered deformable model for capturing poses from cartoon characters. *Joint Symposium on Computational Aesthetics and Sketch-Based Interfaces and Modeling and Non-Photorealistic Animation and Rendering*, 1–12.
- Sven Forstmann and Jun Ohya. 2006. Fast Skeletal Animation by skinned Arc-Spline based Deformation. *Eurographics*, 1–4.
- Tsukasa Fukusato and Akinobu Maejima. 2022. View-Dependent Deformation for 2.5-D Cartoon Models. *Computer Graphics and Applications* 42, 5, 66–75.
- Tsukasa Fukusato, Akinobu Maejima, Takeo Igarashi, and Tatsuo Yotsukura. 2023. Exploring inbetween charts with trajectory-guided sliders for cutout animation. *Multimedia Tools and Applications*, 1–14.
- Tsukasa Fukusato and Shigeo Morishima. 2016. Active comicing for freehand drawing animation. *Mathematical Progress in Expressive Image Synthesis*, 45–56.
- Rinon Gal, Yael Vinker, Yuval Alaluf, Amit H. Bermanto, Daniel Cohen-Or, Ariel Shamir, and Gal Chechik. 2023. Breathing Life Into Sketches Using Text-to-Video Priors. *arXiv preprint arXiv:2311.13608*.
- Songwei Ge, Seungjun Nah, Guilin Liu, Tyler Poon, Andrew Tao, Bryan Catanzaro, David Jacobs, JiaBin Huang, MingYu Liu, and Yogesh Balaji. 2023. Preserve Your

- Own Correlation: A noise prior for video diffusion models. *IEEE International Conference on Computer Vision*, 22930–22941.
- Rohit Girdhar, Mannat Singh, Andrew Brown, Quentin Duval, Samaneh Azadi, Sai S. Rambhatla, Akbar Shah, Xi Yin, Devi Parikh, and Ishan Misra. 2023. Emu Video: Factorizing text-to-video generation by explicit image conditioning. *arXiv preprint arXiv:2311.10709*.
- Yuwei Guo, Ceyuan Yang, Anyi Rao, Yaohui Wang, Yu Qiao, Dahua Lin, and Bo Dai. 2023. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. *arXiv preprint arXiv:2307.04725*.
- Agrim Gupta, Lijun Yu, Kihyuk Sohn, Xiuye Gu, Meera Hahn, FeiFei Li, Irfan Essa, Jiang Lu, and José Lezama. 2023. Photorealistic video generation with diffusion models. *arXiv preprint arXiv:2312.06662*.
- Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P. Kingma, Ben Poole, Mohammad Norouzi, David J. Fleet, and Tim Salimans. 2022. Imagen Video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*.
- Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. 2022. Cogvideo: Large-scale pretraining for text-to-video generation via transformers. *arXiv preprint arXiv:2205.15868*.
- Alexander Hornung, Ellen Dekkers, and Leif Kobbelt. 2007. Character animation from 2D pictures and 3D motion data. *ACM Transactions on Graphics* 26, 1, 1–9.
- Hui Huang, Shihao Wu, Daniel Cohen-Or, Minglun Gong, Hao Zhang, Guiqing Li, and Baoquan Chen. 2013. L1-medial skeleton of point cloud. *ACM Transactions on Graphics* 32, 4, 65–1.
- Zhewei Huang, Tianyuan Zhang, Wen Heng, Boxin Shi, and Shuchang Zhou. 2022. Real-time intermediate flow estimation for video frame interpolation. *European Conference on Computer Vision*, 624–642.
- Takeo Igarashi, Tomer Moscovich, and John F. Hughes. 2005. As-rigid-as-possible shape manipulation. *ACM Transactions on Graphics* 24, 3, 1134–1141.
- Shir Iluz, Yael Vinker, Amir Hertz, Daniel Berio, Daniel Cohen-Or, and Ariel Shamir. 2023. Word-as-Image for Semantic Typography. *arXiv preprint arXiv:2303.01818*.
- Alec Jacobson, Ilya Baran, Jovan Popovic, and Olga Sorkine. 2011. Bounded biharmonic weights for real-time deformation. *ACM Transactions on Graphics* 30, 4, 78.
- Ajay Jain, Amber Xie, and Pieter Abbeel. 2022. VectorFusion: Text-to-SVG by Abstracting Pixel-Based Diffusion Models. *arXiv preprint arXiv:2211.11319*.
- Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz. 2018. Super Slomo: High quality estimation of multiple intermediate frames for video interpolation. *IEEE Conference on Computer Vision and Pattern Recognition*, 9000–9008.
- Tao Jiang, Peng Lu, Li Zhang, Ningsheng Ma, Rui Han, Chengqi Lyu, Yining Li, and Kai Chen. 2023. RTMPose: Real-Time Multi-Person Pose Estimation based on MMPose. *arXiv preprint arXiv:2303.07399*.
- Shizuo Kaji, Sampel Hirose, Shigehiro Sakata, Yoshihiro Mizoguchi, and Ken Anjyo. 2012. Mathematical analysis on affine maps for 2D shape interpolation. *Eurographics Symposium on Computer Animation*, 71–76.
- Ladislav Kavan, Steven Collins, Jivri vZára, and Carol O’Sullivan. 2007. Skinning with dual quaternions. *Symposium on Interactive 3D Graphics and Games*, 39–46.
- Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980*.
- Dan Kondratyuk, Lijun Yu, Xiuye Gu, José Lezama, Jonathan Huang, Rachel Hornung, Hartwig Adam, Hassan Akbari, Yair Alon, Vignesh Birodkar, Yong Cheng, MingChang Chiu, Josh Dillon, Irfan Essa, Agrim Gupta, Meera Hahn, Anja Hauth, David Hendon, Alonso Martinez, David Minnen, David Ross, Grant Schindler, Mikhail Sirotenko, Kihyuk Sohn, Krishna Somandepalli, Huisheng Wang, Jimmy Yan, MingHsuan Yang, Xuan Yang, Bryan Seybold, and Lu Jiang. 2023. Videopoe: A large language model for zero-shot video generation. *arXiv preprint arXiv:2312.14125*.
- Binh H. Le and JP Lewis. 2019. Direct delta mesh skinning and variants. *ACM Transactions on Graphics* 38, 4, 1–13.
- Peizhuo Li, Kfir Aberman, Rana Hanocka, Libin Liu, Olga Sorkine-Hornung, and Baoquan Chen. 2021a. Learning skeletal articulations with neural blend shapes. *ACM Transactions on Graphics* 40, 4, 1–15.
- TzuMao Li, Michal Lukávc, Gharbi Michaël, and Jonathan Ragan-Kelley. 2020. Differentiable Vector Graphics Rasterization for Editing and Learning. *ACM Transactions on Graphics* 39, 6, 1–15.
- Xiaoyu Li, Bo Zhang, Jing Liao, and Pedro V. Sander. 2021b. Deep sketch-guided cartoon video inbetweening. *IEEE Transactions on Visualization and Computer Graphics* 28, 8, 2938–2952.
- Lijuan Liu, Youyi Zheng, Di Tang, Yi Yuan, Changjie Fan, and Kun Zhou. 2019. NeuroSkinning: Automatic skin binding for production characters with deep graph networks. *ACM Transactions on Graphics* 38, 4, 1–12.
- Ziwei Liu, Raymond A. Yeh, Xiaou Tang, Yiming Liu, and Aseem Agarwala. 2017. Video frame synthesis using deep voxel flow. *IEEE International Conference on Computer Vision*, 4463–4471.
- Liyang Luo, Ruizheng Wu, Huajia Lin, Jiangbo Lu, and Jiaya Jia. 2022. Video frame interpolation with transformer. *IEEE Conference on Computer Vision and Pattern Recognition*, 3532–3542.
- Alexander Mathis, Pranav Mamidanna, Kevin M. Cury, Taiga Abe, Venkatesh N. Murthy, Mackenzie W. Mathis, and Matthias Bethge. 2018. DeepLabCut: Markerless pose estimation of user-defined body parts with deep learning. *Nature Neuroscience* 21, 9, 1281–1289.
- Gal Metzer, Elad Richardson, Or Patashnik, Raja Giryes, and Daniel Cohen-Or. 2023. Latent-NeRF for Shape-Guided Generation of 3D Shapes and Textures. *IEEE Conference on Computer Vision and Pattern Recognition*, 12663–12673.
- Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. *European Conference on Computer Vision*.
- XunLong Ng, KianEng Ong, Qichen Zheng, Yun Ni, SiYong Yeo, and Jun Liu. 2022. Animal Kingdom: A large and diverse dataset for animal behavior understanding. *IEEE Conference on Computer Vision and Pattern Recognition*, 19023–19034.
- Bolin Ni, Houwen Peng, Minghao Chen, Songyang Zhang, Gaofeng Meng, Jianlong Fu, Shiming Xiang, and Haibin Ling. 2022. Expanding language-image pretrained models for general video recognition. *European Conference on Computer Vision*, 1–18.
- Haomiao Ni, Changhao Shi, Kai Li, Sharon X. Huang, and Martin Renqiang Min. 2023. Conditional Image-to-Video Generation with Latent Flow Diffusion Models. *IEEE Conference on Computer Vision and Pattern Recognition*, 18444–18455.
- Simon Niklaus and Feng Liu. 2018. Context-aware synthesis for video frame interpolation. *IEEE Conference on Computer Vision and Pattern Recognition*, 1701–1710.
- Simon Niklaus and Feng Liu. 2020. Softmax splatting for video frame interpolation. *IEEE International Conference on Computer Vision*, 5437–5446.
- Simon Niklaus, Long Mai, and Feng Liu. 2017a. Video frame interpolation via adaptive convolution. *IEEE Conference on Computer Vision and Pattern Recognition*, 670–679.
- Simon Niklaus, Long Mai, and Feng Liu. 2017b. Video frame interpolation via adaptive separable convolution. *IEEE International Conference on Computer Vision*, 261–270.
- Junheum Park, Keonsoo Ko, Chul Lee, and Chang-Su Kim. 2020. BMBC: Bilateral motion estimation with bilateral cost volume for video interpolation. *European Conference on Computer Vision*, 109–125.
- Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. 2022. Dreamfusion: Text-to-3D using 2D Diffusion. *arXiv preprint arXiv:2209.14988*.
- Zhiyu Qu, Tao Xiang, and YiZhe Song. 2023. SketchDreamer: Interactive Text-Augmented Creative Sketch Ideation. *arXiv preprint arXiv:2308.14191*.
- Alec Radford, Jong-Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. *International Conference on Machine Learning*, 8748–8763.
- Fitsum Reda, Janne Kontkanen, Eric Tabellion, Deqing Sun, Caroline Pantofaru, and Brian Curless. 2022. FILM: Frame interpolation for large motion. *European Conference on Computer Vision*, 250–266.
- Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-Resolution Image Synthesis with Latent Diffusion Models. *IEEE Conference on Computer Vision and Pattern Recognition*, 10684–10695.
- Jonathan Richard Shewchuk. 1996. Triangle: Engineering a 2D quality mesh generator and Delaunay triangulator. *Workshop on Applied Computational Geometry*, 203–222.
- Hyeonjun Sim, Jiyoung Oh, and Munchul Kim. 2021. XVFI: Extreme video frame interpolation. *IEEE International Conference on Computer Vision*, 14489–14498.
- Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman. 2022. Make-a-Video: Text-to-Video generation without text-video data. *arXiv preprint arXiv:2209.14792*.
- Li Siyao, Tianpei Gu, Weiye Xiao, Henghui Ding, Ziwei Liu, and ChangeLoy Chen. 2023. Deep Geometrized Cartoon Line Inbetweening. *IEEE International Conference on Computer Vision*, 7291–7300.
- Li Siyao, Shiyu Zhao, Weijiang Yu, Wenxiu Sun, Dimitris Metaxas, ChangeLoy Chen, and Ziwei Liu. 2021. Deep animation video interpolation in the wild. *IEEE Conference on Computer Vision and Pattern Recognition*, 6587–6595.
- Harrison J. Smith, Qingyuan Zheng, Yifei Li, Somya Jain, and Jessica K. Hodgins. 2023. A Method for Animating Children’s Drawings of the Human Figure. *ACM Transactions on Graphics* 42, 3, 1–15.
- Qingkun Su, Xue Bai, Hongbo Fu, ChiewLan Tai, and Jue Wang. 2018. Live Sketch: Video-Driven dynamic deformation of static drawings. *Conference on Human Factors in Computing Systems*, 1–12.
- Meiqi Sun, Zhonghan Zhao, Wenhao Chai, Hanjun Luo, Shidong Cao, Yanting Zhang, JenqNeng Hwang, and Gaoang Wang. 2023. UniAP: Towards Universal Animal Perception in Vision via Few-shot Learning. *arXiv preprint arXiv:2308.09953*.
- Andrea Tagliasacchi, Ibraheem Alhashim, Matt Olson, and Hao Zhang. 2012. Mean curvature skeletons. *Computer Graphics Forum* 31, 5, 1735–1744.
- Maham Tanveer, Yizhi Wang, Ali Mahdavi-Amiri, and Hao Zhang. 2023. DS-Fusion: Artistic Typography via Discriminated and Stylized Diffusion. *arXiv preprint arXiv:2303.09604*.
- Maham Tanveer, Yizhi Wang, Ruiqi Wang, Nanxuan Zhao, Ali Mahdavi-Amiri, and Hao Zhang. 2024. AnaMoDiff: 2D Analogical Motion Diffusion via Disentangled

- Denoising. *arXiv preprint arXiv:2402.03549*.
- Christina Tsalicoglou, Fabian Manhardt, Alessio Tonioni, Michael Niemeyer, and Federico Tombari. 2023. TextMesh: Generation of Realistic 3D Meshes From Text Prompts. *arXiv preprint arXiv:2304.12439*.
- Ruben Villegas, Mohammad Babaeizadeh, PieterJan Kindermans, Hernan Moraldo, Han Zhang, Mohammad T. Saffar, Santiago Castro, Julius Kunze, and Dumitru Erhan. 2022. Phenaki: Variable length video generation from open domain textual description. *arXiv preprint arXiv:2210.02399*.
- Jiuniu Wang, Hangjie Yuan, Dayou Chen, Yingya Zhang, Xiang Wang, and Shiwei Zhang. 2023a. Modelscope text-to-video technical report. *arXiv preprint arXiv:2308.06571*.
- Xiang Wang, Hangjie Yuan, Shiwei Zhang, Dayou Chen, Jiuniu Wang, Yingya Zhang, Yujun Shen, Deli Zhao, and Jingren Zhou. 2023b. VideoComposer: Compositional Video Synthesis with Motion Controllability. *arXiv preprint arXiv:2306.02018*.
- Brian Whited, Gioacchino Noris, Maryann Simmons, Robert W. Sumner, Markus Gross, and Jarek Rossignac. 2010. Betweenit: An interactive tool for tight inbetweening. *Computer Graphics Forum* 29, 2, 605–614.
- Nora S. Willett, Hujung V. Shin, Zeyu Jin, Wilmot Li, and Adam Finkelstein. 2020. Pose2Pose: Pose selection and transfer for 2D character animation. *International Conference on Intelligent User Interfaces*, 88–99.
- Chenfei Wu, Jian Liang, Lei Ji, Fan Yang, Yuejian Fang, Daxin Jiang, and Nan Duan. 2022. Nüwa: Visual synthesis pre-training for neural visual world creation. *European Conference on Computer Vision*, 720–736.
- Jinbo Xing, Menghan Xia, Yong Zhang, Haoxin Chen, Xintao Wang, TienTsin Wong, and Ying Shan. 2023b. Dynamicrafter: Animating open-domain images with video diffusion priors. *arXiv preprint arXiv:2310.12190*.
- Ximing Xing, Chuang Wang, Haitao Zhou, Jing Zhang, Qian Yu, and Dong Xu. 2023a. DiffSketcher: Text Guided Vector Sketch Synthesis through Latent Diffusion Models. *arXiv preprint arXiv:2306.14685*.
- Xiangyu Xu, Li Siyao, Wenxiu Sun, Qian Yin, and Ming-Hsuan Yang. 2019. Quadratic video interpolation. *Advances in Neural Information Processing Systems* 32.
- Yufei Xu, Jing Zhang, Qiming Zhang, and Dacheng Tao. 2022. Vitpose: Simple vision transformer baselines for human pose estimation. *Advances in Neural Information Processing Systems* 35, 38571–38584.
- Zhan Xu, Yang Zhou, Evangelos Kalogerakis, Chris Landreth, and Karan Singh. 2020. RigNet: Neural rigging for articulated characters. *arXiv preprint arXiv:2005.00559*.
- Jie Yang, Bingliang Li, Fengyu Yang, Ailing Zeng, Lei Zhang, and Ruimao Zhang. 2023a. Boosting Human-Object Interaction Detection with Text-to-Image Diffusion Model. *arXiv preprint arXiv:2305.12252*.
- Jie Yang, Ailing Zeng, Ruimao Zhang, and Lei Zhang. 2023b. UniPose: Detecting Any Keypoints. *arXiv preprint arXiv:2310.08530*.
- Shaokai Ye, Anastasiia Filippova, Jessy Lauer, Maxime Vidal, Steffen Schneider, Tian Qiu, Alexander Mathis, and Mackenzie W. Mathis. 2022. SuperAnimal models pretrained for plug-and-play analysis of animal behavior. *arXiv preprint arXiv:2203.07436*.
- Xin Yuan, Jinoo Baek, Keyang Xu, Omer Tov, and Hongliang Fei. 2024. Inflation with Diffusion: Efficient Temporal Adaptation for Text-to-Video Super-Resolution. *IEEE Winter Conference on Applications of Computer Vision*, 489–496.
- Shiwei Zhang, Jiayu Wang, Yingya Zhang, Kang Zhao, Hangjie Yuan, Zhiwu Qin, Xiang Wang, Deli Zhao, and Jingren Zhou. 2023. I2vgen-xl: High-Quality image-to-video synthesis via cascaded diffusion models. *arXiv preprint arXiv:2311.04145*.