AniDoc: Animation Creation Made Easier

Yihao Meng[1,2] Hao Ouyang[2] Hanlin Wang[3,2] Qiuyu Wang[2] Wen Wang[4,2]
Ka Leong Cheng[1,2] Zhiheng Liu[5] Yujun Shen[2] Huamin Qu[†,1]

[1] HKUST [2] Ant Group [3] NJU [4] ZJU [5] HKU

Figure 1. `AniDoc` colorizes a sequence of sketches based on a character design reference with high fidelity, even when the sketches significantly differ in pose and scale. Additionally, the model supports sparse sketch inputs, enabling effective interpolation and high-quality colorization simultaneously, as shown in the last row.

Abstract

The production of 2D animation follows an industry-standard workflow, encompassing four essential stages: character design, keyframe animation, in-betweening, and coloring. Our research focuses on reducing the labor costs in the above process by harnessing the potential of increasingly powerful generative AI. Using video diffusion models as the foundation, `AniDoc`[1] emerges as a video line art colorization tool, which automatically converts sketch sequences into colored animations following the reference character specification. Our model exploits correspondence matching as an explicit guidance, yielding strong robustness to the variations (e.g., posture) between the reference character and each line art frame. In addition, our model could even automate the in-betweening process, such that users can easily create a temporally consistent animation by simply providing a character image as well as the start and end sketches. Our code is available at: https://yihao-meng.github.io/AniDoc_demo.

[1] "Doc" is one of the seven dwarfs in *Snow White and the Seven Dwarfs*, the first animated feature film produced by Disney.

1. Introduction

The animation industry, particularly in the realm of 2D anime production, relies heavily on the meticulous process of coloring line art to bring characters and scenes to life. Colorization of line art [11, 12, 17, 33, 42, 51, 65] in videos is a critical task that not only adds aesthetic value but also enhances the storytelling experience by conveying emotions and actions vividly. Automating this process holds signifi-

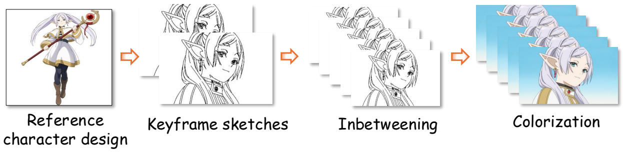

Figure 2. Illustration of the workflow of 2D animation production.

cant potential to streamline production workflows, reduce labor costs, and accelerate content creation, meeting the growing demand for high-quality animated content.

In the current anime production pipeline, artists typically begin with character design sheets that define the visual attributes of characters. These designs are then translated into keyframe sketches — crucial frames that outline the primary poses and movements in a scene. Next, artists create in-betweening sketches, which are the frames drawn between the keyframes to define the detailed movement and transition [40, 63]. Traditionally, these frames are manually colored, a time-consuming task that involves careful attention to ensure consistency with the original character designs. Fig. 2 illustrates each step of this pipeline. Our work aligns seamlessly with this pipeline, aiming to automate the colorization process while maintaining fidelity to the original character designs and ensuring temporal consistency across frames.

However, automating line art colorization [25, 57] presents several challenges. One primary difficulty lies in the mismatch between the character design and the line art sketches, where the angles, proportions, and poses in the design may not align with those in the keyframe sketches. Additionally, achieving temporal consistency is crucial; colorizing each frame individually can lead to flickering or inconsistencies, detracting from the viewer's experience [5, 31, 62]. Previous approaches [18, 44, 54] have attempted to address these challenges but with limitations. They often assume the availability of colorized versions of keyframes and rely on dense line art guidance. This assumption significantly increases the workload on artists, as it requires manual coloring of multiple keyframes and detailed line art inputs, making the process tedious and labor-intensive. Moreover, some methods suffer from color information leakage due to their training pipelines. Specifically, they use non-binarized sketches extracted from color images using neural networks for training, unintentionally incorporating color information from the original images into the sketches. This information leakage undermines the practicality of these methods, as real-world sketches do not contain such implicit color information—a concern we analyze further in our methodology.

To overcome these challenges, we propose a novel all-in-one model that streamlines the colorization process within a single framework. Our model leverages the priors from pretrained diffusion-based video generation models [1, 36], harnessing their ability to capture temporal dynamics and visual coherence. The key designs of our approach are as follows: **Correspondence-guided Colorization**: We address the misalignment between the reference character design and the input line art sketches by incorporating an explicit correspondence mechanism. The injection module is designed to integrate color and style information from the reference into the line art, effectively improving color accuracy and consistency. **Binarization and Background Augmentation**: To reflect real usage scenarios, we binarize the condition sketches, forcing the model to truly learn to extract color information from the reference character design, rather than relying on recovering color information leaked from the non-binarized sketches. This constraint poses additional challenges for the model to accurately colorize the line art. To mitigate the instability during training due to this reduced information, we incorporate background augmentation strategy which greatly improve the colorization result. **Sparse Sketch Training**: Our model employs a two-stage training strategy that first learns the colorization ability and then removes the intermediate sketches to learn the interpolation ability. By learning the interpolation between keyframes, our model maintains temporal consistency without extensive human intervention.

Our method demonstrates superior performance both quantitatively and qualitatively compared to existing approaches. It effectively colorizes line art sketches in videos, maintaining high fidelity to the reference character designs and ensuring temporal consistency across frames. Moreover, we demonstrate that a single reference character image can be used to colorize sketches from different segments featuring the same character, even when these sketches differ significantly in scale, pose, and action from the reference design. Our work represents a significant step toward automated, efficient, and artistically consistent animation production, with potential applications extending beyond anime to various forms of digital art and media.

2. Related Work

2.1. Line Art Image Colorization

Line art colorization [2, 10, 22, 27, 28, 37, 56–58, 61] differs from natural image colorization, as it lacks an illuminance channel and only contains structural information, offering more flexibility. Traditional methods [35, 41] rely on users manually adding color to specific regions. The advent of deep learning has advanced this field, with techniques like color hint points [55], color scribbles [8], text tags [23], and natural language prompts [64]. Reference-based colorization, where users provide a reference image to guide the coloring, has also gained popularity. Methods like [38] use segmented graphs, while Chen et al. [6] employ active learning, and [3] apply attention networks. AnimeDiffusion [4] introduces the first diffusion-based reference-based framework for anime face colorization. However, these methods colorize sketch images separately, struggling with temporal coherence when colorize a sequence of sketches.

2.2. Reference-based Line Art Video Colorization

Several approaches extend reference-based colorization to videos. Given an input of reference image, these methods colorize the corresponding sketch sequence based on the reference image's color information. LCMFTN [60] trains a model using animation frame pairs but lacked temporal coherence. TCVC [44] uses previously colorized frames to maintain short-term consistency, but errors accumulate over time. TRE-Net [46] mitigates this by using both the first frame and previously generated frames. ACOF [54] propagates colors based on optical flow but requires refinement. The most recent work LVCD [18] proposes the first video diffusion model to colorize sketch video, but suffer from the issue of non-binarized sketch information leakage, limiting its real-world applicability. Most importantly, all of these methods require the color version of the first frame of each video clip, while in the actual anime production workflow, colorists typically only receive the character design image and need to use that image to colorize different clips featuring the character.

2.3. Video Interpolation

With the rapid development of controllable video generation models [30, 50], video interpolation has achieved increasingly better results. Unlike reference-based colorization, video interpolation [9, 16, 19, 20, 24, 39, 47] aims to generate in-between frames from both the first and last frames. SparseCtrl [13] and SEINE [7] extend video interpolation using pretrained text-to-video diffusion models and additional image conditions. Shi et al. [39] adapt video interpolation for cartoon animation with temporal constraint networks. AnimeInterp [40] interpolates middle frames by warping with predicted flows. ToonCrafter [49] introduces a diffusion-based video interpolation model. Zhu et al. [63] proposes a thin-plate spline-based interpolation module for animation sketch inbetweening. We aim at an all-in-one model that simultaneously performs automatic interpolation and colorization for anime. However, more precise interpolation methods [47], can be integrated to achieve enhanced control over sketch manipulation.

3. Method

We formally define the problem of line art video colorization with reference images. Given a reference image I_{ref} that encapsulates the desired color and style of the character and a sequence of binarized line art sketches $\{S_t\}_{t=1}^{T}$, where S_t is the sketch at time frame t, our objective is to generate a sequence of colorized frames $\{I_t\}_{t=1}^{T}$ such that:

1. Each frame I_t is a colorized version of the sketch S_t.
2. The colorization is consistent with the character design.
3. The sequence $\{I_t\}$ is temporally coherent, ensuring smooth transitions without flickering artifacts.

Formally, we aim to learn a function f that maps the sketches and the reference image to the colorized frames:

$$\{I_t\}_{t=1}^{T} = f(\{S_t\}_{t=1}^{T}, I_{\text{ref}}). \tag{1}$$

3.1. Motivation and Pipeline Design

We summarize the shortcomings of state-of-the-art approaches when handling real-world animation production scenarios and design modules to overcome them.

Mismatch Between Character Design and Input Sketches. Existing methods rely on the assumption that the reference image is strongly pixel-aligned with the first frame of the sketch sequence. When the reference provided is not the first-frame image, but instead a character design from a different angle, the model struggles to correctly match the colors and details, as shown in Fig. 5. This limitation necessitates manual coloring for each clip's keyframes in the animation, which is labor-intensive and counteracts the benefits of automation. To address this, we aligns and transfers color information from reference character design to input sketches with correspondence-guided control module, as described in Sec. 3.2.

Degradation with Binarized Sketches. Previous methods often rely on sketches extracted from color images using learned neural networks. These non-binarized sketches contain unintended color information leaks from the original color images, although invisible to the human eye. When training with these non-binarized sketches, the model tends to learn how to recover these hidden color information rather than correctly learning to find the corresponding parts in the reference to color the sketch, resulting in models that perform poorly when applied to binarized sketches. As depicted in Fig. 4, when previous methods are applied to binarized sketches, the outputs suffer from severe degradation. The colorization is inaccurate, and the visual quality is significantly reduced compared to when non-binarized sketches are used. To address this issue, we mirror the real production conditions by adopting the binary sketch in training and apply background augmentation to enhance the robustness as in Sec. 3.3.

Reliance on Dense Sketches as Conditions. Previous methods often require dense sketches to maintain temporal consistency. Manually drawing inbetweening sketches in an animation is costly. We aim to use only S_1, S_T to further improve the scalability of automatic creation and propose the sparse sketch training scheme in Sec. 3.4.

Pipeline Design. Following Stable Video Diffusion (SVD) [1], our main architecture consists of a denoising 3D U-Net designed for video generation. The reference

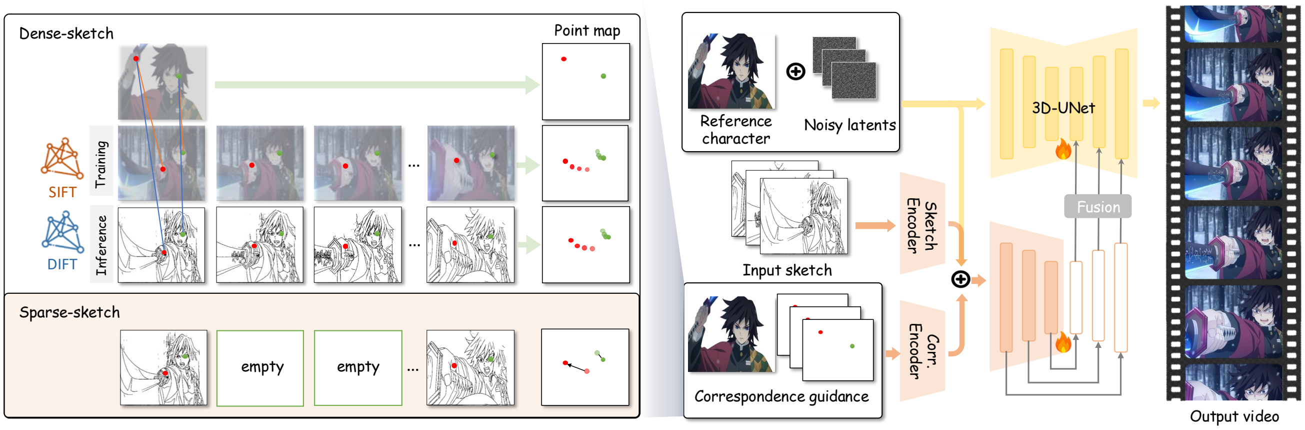

Figure 3. Overview of `AniDoc` pipeline. We adopt a two-stage training strategy. In the *dense-sketch* training stage, we explicitly extract matching keypoints pairs between the reference image and each frame of the training video, constructing point maps to represent the correspondences. In the *sparse-sketch* training stage, we remove the intermediate frame sketches and use the matching points from the start and end frames to interpolate point trajectories, which guide the generation of the intermediate frames.

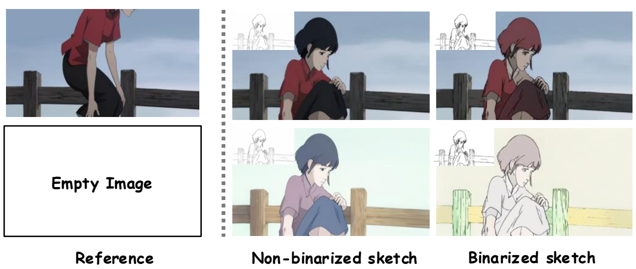

Figure 4. Illustration of color leakage issue in non-binarized sketch. For previous video colorization method [18], when given non-binarized sketch, even if the reference is an empty image, it can still generate colorized results with similar color pattern to the ground truth. After binarizing the sketch, the colorization results degrade significantly.

image latent is duplicated across the number of frames and concatenated along the channel dimension with the noisy latent so that the reference image information can be integrated into the colorization process through multiple self-attention layers in the 3D U-Net encoder. To inject the correspondence between the reference character design and the sketch, we explicitly extract corresponded keypoints and construct point maps. The correspondence information and sketch information are then encoded by a 3D U-Net dual branch similar to [59] , and injected into the main branch as control signals. To construct training pairs of reference character designs and corresponding sketch sequences, we select long videos from the Sakuga-42M dataset [32], where the characters naturally undergo rich transformations, including changes in position, posture, angle, scale, etc., which align with the needs of our model.

The training objective for our model is as follows:

$$\mathcal{L} = \mathbb{E}_{z_t, z^0, t, \epsilon}\left[\left\|\epsilon - \epsilon_\theta^c\left(z_t; t, z^0, c_{sketch}, c_{corr}\right)\right\|^2\right], \quad (2)$$

where z^0 denotes VAE-encoded latent feature of the reference image, c_{sketch} means the sketch control signal, c_{corr} means the correspondence control signal, and ϵ_θ^c is the combination of the denoising U-Net and the control branch.

3.2. Correspondence-guided Colorization

During training, we use an off-the-shelf keypoint matching method LightGlue [26] with SIFT descriptor [29] to extract matching keypoints between the reference image and the first frame of the training video. The matched keypoints are denoted as $\left\{(x^i_{\text{ref}}, y^i_{\text{ref}}), (x^i_1, y^i_1)\right\}_{i=1}^n$, where n refers to the number of matching pairs. Based on these matched keypoints, we construct a point map pair $P_1 = (P_{\text{ref}}, P_{1,\text{sketch}})$, representing the matching correspondence of the reference image and the first frame. Each point map is in the size of $H \times W$ (same as the image resolution), where the coordinates corresponding to the matched keypoints are marked with the same integer label:

$$P_{\text{ref}}(x^i_{\text{ref}}, y^i_{\text{ref}}) = P_{1,\text{sketch}}(x^i_1, y^i_1) = i. \quad (3)$$

For unmatched pixels, the value is set to 0.

We then employ Co-Tracker [21] to track the movement of keypoints $\left\{(x^i_1, y^i_1)\right\}_{i=1}^n$ in each frame and construct correspondence point maps in the same way. As a result, we obtain a point map $P_{\text{seq}} = \{P_t\}_{t=1}^T \in \mathbb{R}^{2\times T\times H\times W}$, which explicitly encodes the correspondence between the reference image and each sketch in the training video.

We concatenate the point map P_{seq} with the reference image I_{ref} together as explicit correspondence guidance information and integrate it into the control branch. Specifically, for a given keypoint (x_t^i, y_t^i) in sketch frame S_t, the model can use the point map P_t to obtain the corresponding location $(x_{\text{ref}}^i, y_{\text{ref}}^i)$ in the reference image I_{ref} and extract corresponding color information at that location. Thus, our video generation process can be represented as:

$$\{I_t\}_{t=1}^T = D(I_{\text{ref}}, E(\{S_t\}_{t=1}^T, \{P_t\}_{t=1}^T, I_{\text{ref}})), \quad (4)$$

where D refers to the denoising process of the 3D diffusion U-net, E refers to the control branch encoder.

Semantic Keypoint Matching During Inference. During training, we apply LightGlue [26] with SIFT descriptor [29] for keypoint selection and matching between the reference image and the training video frames due to its fast speed. However, during inference, we do not have the ground truth color image to extract corresponding keypoints. Methods like the SIFT descriptor, which are focusing on low-level image feature, fail to correctly match keypoints between the color reference image and the sketch due to the large domain gap. One recent work DIFT [43] have found that features extracted by diffusion models can achieve semantic-level matching. Thus, during inference, we first extract keypoints in the reference character design using X-Pose [52], and find the matching keypoints in the given sketch using semantic feature DIFT.

3.3. Binarization and Background Augmentation

In real animation production, sketches provided to colorists are typically binarized line drawings without grayscale shading or hidden color information. To simulate this real-world condition, we apply a binarization process to the extracted sketches during training by setting pixel values greater than 200 to 255 and others to 0. This converts the sketch S_{raw} into a binary image S_{bin}, with black lines (0) and white background (255).

Using binarized sketches as conditioning inputs poses significant challenges for the model. One primary issue is the ambiguity between the background and large white regions in the foreground. Both are represented by the same pixel value (255), making it difficult for the model to distinguish. This ambiguity can lead to confusion during colorization, resulting in erroneous outputs where the model may incorrectly color background regions or fail to accurately color foreground elements. Such failure cases are evident in our ablation studies (refer to Sec. 4.4).

To address this problem, we enhance the training process through background augmentation. Specifically, we randomly remove the background of the reference image with a 50% probability during training process, using an off-the-shelf background removal model [34]. This forces the model to learn to distinguish between the foreground and background regions. In the foreground region, the model learns to extract color information from the correct areas of the reference character design. In the background region, the model is required to rely more heavily on its internal generative prior, enabling it to produce a background that is coherent with the foreground character.

3.4. Sparse Sketch Training

To further reduce the necessity of drawing intermediate sketches for animations with simple motions, we introduce a sparse sketch training strategy in a two stage way. In the first stage, the training is conducted with all frame sketches available. We hope that the model learns how to correctly extract information from the point map in this stage, which will guide the training in the next stage.

After completing the first stage training, we remove the sketch condition for intermediate frames and use keypoints information to guide the interpolation. Inspired by [48], we additionally transform the keypoint coordinates into a Gaussian heatmap G_t which is more suitable for trajectory control, and concatenate G_t with the original point map P_t together. Specifically, the model is conditioned on the start and end sketches, S_1 and S_T, as well as the pointmap $P_t \oplus G_t$ at each frame:

$$\{I_t\}_{t=1}^T = D(I_{\text{ref}}, E(S_1, S_T, \{P_t \oplus G_t\}_{t=1}^T, I_{\text{ref}})). \quad (5)$$

Noted that during training the point trajectories are obtained through point tracking in the training video using Co-Tracker [21], while during inference we extract matching keypoints pairs between the start and end sketches and linearly interpolate the intermediate point trajectories.

In this sparse-sketch training stage, we randomly select up to 5 keypoints, corresponding to 5 trajectories. The selection probability is determined by the magnitude of the motion, with trajectories having larger motion being more likely to be selected. Guided by the keypoint trajectories, our model can produce smooth intermediate colorized frames with only sparse sketch inputs.

4. Experiments

4.1. Implementation Details

Our method is built upon SVD and is trained on the Sakuga-42M [32] dataset, which comprises a large number of anime clips with diverse styles. To create reference images and sketch videos with large differences, we exclude clips with fewer than 50 frames, ultimately retaining around 150k video clips. We set the interval between the reference image and the first frame of the target video to 32 frames, with the target video length being 14 frames. During the first training stage, where the generation is conditioned on per-frame sketches, we simultaneously fine-tune all parameters of both the U-Net and ControlNet, including both the spatial and temporal attention layers. This is done for 100k steps using the AdamW optimizer with a learning rate

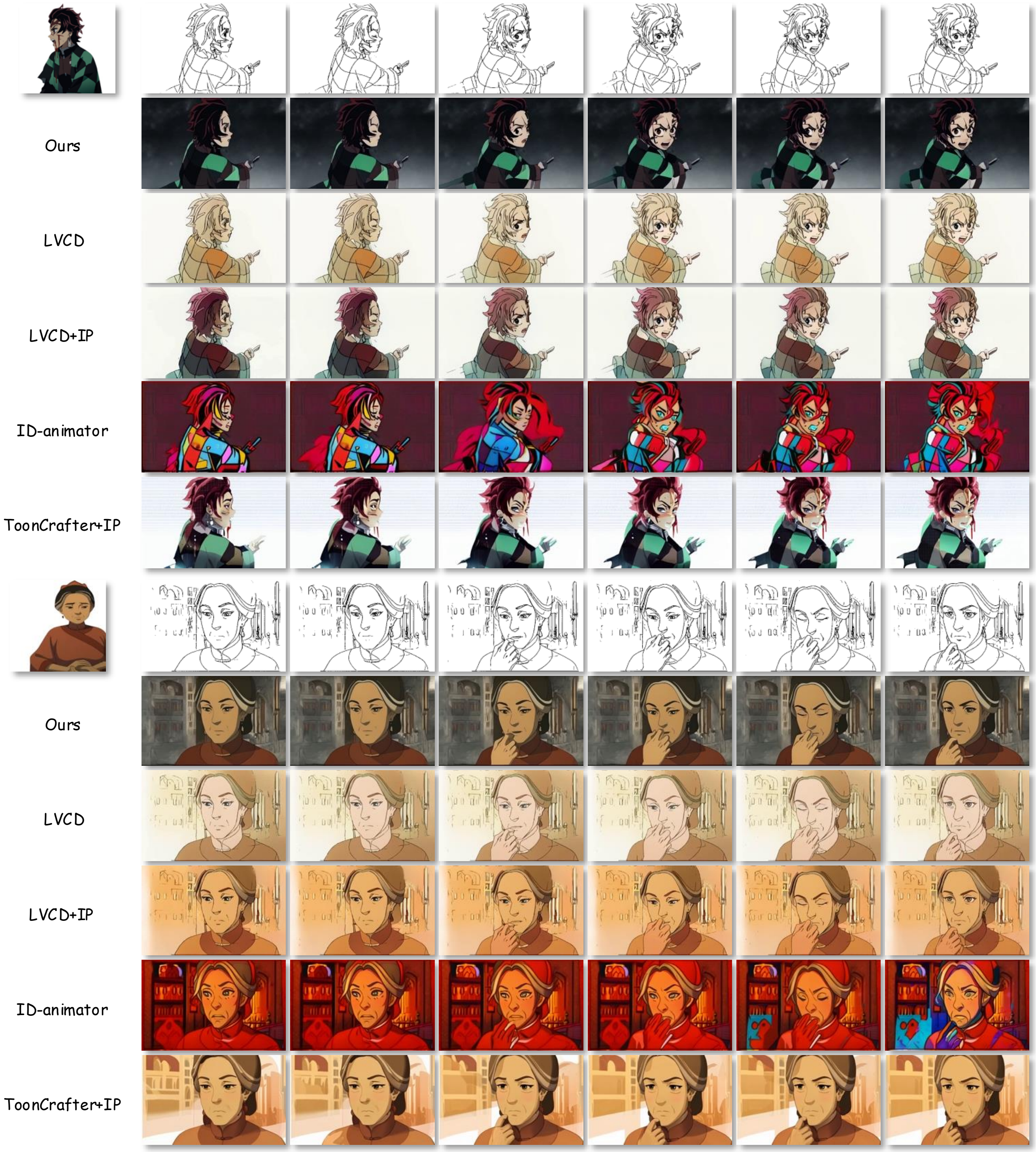

Figure 5. Visual comparison of reference-based colorization with four methods LVCD [18], LVCD+IP-Adapter [53], ID-animator [14], ToonCrafter [49].

of 1×10^{-5}, at a resolution of 256×256 due to GPU memory constraints. Subsequently, we freeze other layers and fine-tune the spatial layers for an additional 10k steps at a resolution of 512×320. We randomly select up to 50 keypoints to enhance the model's robustness to varying numbers of keypoints. During the sparse sketch training

Figure 6. Ablations on each component. "*w/o* matching" indicates without the corresponding matching module, "*w/o* binarize" indicates without binarization and background augmentation.

stage, we remove the middle frames' sketches and further fine-tuned all parameters for 100k steps. The training is conducted on 16 NVIDIA A100 GPUs with a total batch size of 16, and the entire training process takes 5 days.

4.2. Comparison

To comprehensively evaluate the coloring ability of our model, we randomly select 200 clips from 10 different eras and styles of anime to construct the test set. We select corresponding character design images (without background) as reference images.

We compare our proposed method with two state-of-the-art reference-based line art video colorization frameworks: LVCD [18] and ToonCrafter [49], both of which are based on video diffusion models. Since LVCD's original setting requires the colorized version of the first sketch as a reference, we also compare it with the IP-Adapter [53] + LVCD version. In this case, we first use a diffusion-based image colorization method IP-Adapter to colorize the first frame's sketch, and then use this colorized image as the reference for LVCD. For ToonCrafter, which requires color versions of both the start and end frames, we first use IP-Adapter to colorize the sketches of both frames. Additionally, we select a recent video personalization method, ID-animator [14], which excels at identity preservation in general domains and can achieve a similar function to colorization when combined with ControlNet [59].

Qualitative Comparison. As shown in Fig. 5, our method produces significantly clearer textures and better preserves the character's identity. It performs especially well in scenarios with substantial differences between the reference character design and input sketches, where LVCD and ID-Animator fail to accurately colorize the sketches. Even when providing the colorized first frame using IP-Adapter for these baselines, our method still outperforms them in both visual quality and identity preservation.

Quantitative Comparison. We evaluate the quality of the colorized animations in two aspects: 1). Frame and Video Quality: we adopt Frechet Inception Distance [15] (FID) to measure image quality and assess video quality using Frechet Video Distance [45] (FVD). 2). Colorization Accuracy: we measure the similarity of the colorized frames and the original animation frames using reconstruction metrics including PSNR, SSIM, and LPIPS. For all the metrics, we resize the frames to 256×256 and remove the background for every frame because the reference character design does not include background information.

Table 1. Quantitative comparison with existing baselines on reconstruction and generative metrics.

Method	PSNR↑	SSIM↑	LPIPS↓	FID↓	FVD↓
ID-Animator	15.61	0.3129	0.5151	158.16	677.61
LVCD	15.77	0.6446	0.2580	121.98	584.33
LVCD + IP	16.52	0.6404	0.2639	118.28	496.45
ToonCrafter + IP	14.97	0.3983	0.4532	110.48	492.10
AniDoc (Ours)	**19.23**	**0.7720**	**0.1704**	**54.33**	**230.18**

Table 2. Ablations on correspondence matching module and *binarization + background* augmentation.

Settings	PSNR↑	SSIM↑	LPIPS↓	FID↓	FVD↓
w/o binary	15.41	0.6639	0.2471	94.89	503.23
w/o matching	17.80	0.7068	0.2252	75.91	273.86
AniDoc (Ours)	**19.23**	**0.7720**	**0.1704**	**54.33**	**230.18**

As shown in Tab. 1, our method achieves the best scores across all metrics, indicating high-quality colorization results. For a more comprehensive assessment, we recommend reviewing the supplementary comparison videos.

4.3. Flexible Usage

We assess the flexibility of our model in three distinct settings, as depicted in Fig. 7.

Same Reference with Varying Sketches. By using the same reference, our model is able to generate consistent colorizations across different video clips, even when the sketches differ significantly in terms of pose or scale.

Same Sketch with Different References. When applying different reference images to the same sketch sequence, our method preserves the identity of the character while adapting the finer details, such as lighting and background, according to the distinct styles of the references.

Sparse Input Sketches. Thanks to our two-stage training strategy, AniDoc supports animation with sparse sketches. By using only the start and end sketches, the model effectively produces smooth and coherent animations.

4.4. Ablation Study

We perform ablation studies Tab. 2 on two key components.

Effect of Correspondence Matching. Without the correspondence matching module, the model struggles to

Figure 7. Illustration of the flexible usage of our model. Figure (a) shows the ability of using same reference to colorize different sketches. Figure (b) demonstrates the robustness to different references. Figure (c) shows the sparse-sketch generation results.

localize and transfer detailed color information accurately. As shown in the Fig. 6, the character's black sideburns have not been colored correctly, and the pink color of the hair has mistakenly been applied to the clothing. Incorporating correspondence matching ensures precise alignment between the reference and the input sketches, significantly improving color accuracy.

Effect of Background Augmentation. Without background augmentation during training, the model struggles to distinguish between the foreground (character) and the background. Consequently, it may generate frames where certain regions are incorrectly colorized as pure white or contain artifacts, due to the limited information provided by the binarized sketches. Incorporating background augmentation helps the model to better understand scene context, leading to more accurate and visually pleasing results.

5. Conclusion

In this paper, we introduced a novel all-in-one model for automatic lineart video colorization that integrates seamlessly with existing anime production pipelines. Our approach tackles key challenges such as misalignment between character designs, limited information in binarized sketches and situations with only sparse sketches. Comprehensive experiments demonstrate that our method outperforms state-of-the-art baselines in both quality and temporal consistency. For future work, we aim to incorporate interactive point control to handle subtle color variations and develop stronger, more efficient video models to support longer and higher-quality animation creation, further enhancing the efficiency and creativity of animation production.

Acknowledgements: This work was supported by Ant Group Research Intern Program.

References

[1] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, Varun Jampani, and Robin Rombach. Stable video diffusion: Scaling latent video diffusion models to large datasets. *CoRR*, abs/2311.15127, 2023. 2, 3

[2] Ruizhi Cao, Haoran Mo, and Chengying Gao. Line art colorization based on explicit region segmentation. In *Computer Graphics Forum*, pages 1–10, 2021. 2

[3] Yu Cao, Hao Tian, and P. Y. Mok. Attention-aware anime line drawing colorization. In *IEEE International Conference on Multimedia and Expo, ICME 2023, Brisbane, Australia, July 10-14, 2023*, pages 1637–1642. IEEE, 2023. 2

[4] Yu Cao, Xiangqiao Meng, PY Mok, Tong-Yee Lee, Xueting Liu, and Ping Li. Animediffusion: anime diffusion colorization. *IEEE Transactions on Visualization and Computer Graphics*, 2024. 3

[5] Hernan Carrillo, Michaël Clément, Aurélie Bugeau, and Edgar Simo-Serra. Diffusart: Enhancing line art colorization with conditional diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3486–3490, 2023. 2

[6] Shu-Yu Chen, Jia-Qi Zhang, Lin Gao, Yue He, Shihong Xia, Min Shi, and Fang-Lue Zhang. Active colorization for cartoon line drawings. *IEEE Transactions on Visualization and Computer Graphics*, 28(2):1198–1208, 2020. 2

[7] Xinyuan Chen, Yaohui Wang, Lingjun Zhang, Shaobin Zhuang, Xin Ma, Jiashuo Yu, Yali Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. Seine: Short-to-long video diffusion model for generative transition and prediction. In *The Twelfth International Conference on Learning Representations*, 2023. 3

[8] Yuanzheng Ci, Xinzhu Ma, Zhihui Wang, Haojie Li, and Zhongxuan Luo. User-guided deep anime line art colorization with conditional adversarial networks. In *Proceedings of the 26th ACM international conference on Multimedia*, 2018. 2

[9] Duolikun Danier, Fan Zhang, and David Bull. LDMVFI: video frame interpolation with latent diffusion models. In *Assoc. Adv. Artif. Intell.*, 2024. 3

[10] Zhi Dou, Ning Wang, Baopu Li, Zhihui Wang, Haojie Li, and Bin Liu. Dual color space guided sketch colorization. *IEEE Trans. Image Process.*, 30:7292–7304, 2021. 2

[11] Sébastien Fourey, David Tschumperlé, and David Revoy. A fast and efficient semi-guided algorithm for flat coloring line-arts. *Le Centre pour la Communication Scientifique Directe - HAL - Archive ouverte HAL,Le Centre pour la Communication Scientifique Directe - HAL - Archive ouverte HAL*, 2018. 1

[12] Chie Furusawa, Kazuyuki Hiroshiba, Keisuke Ogaki, and Yuri Odagiri. Comicolorization: Semi-automatic manga colorization. *Cornell University - arXiv,Cornell University - arXiv*, 2017. 1

[13] Yuwei Guo, Ceyuan Yang, Anyi Rao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Sparsectrl: Adding sparse controls to text-to-video diffusion models. In *Eur. Conf. Comput. Vis.*, pages 330–348, 2024. 3

[14] Xuanhua He, Quande Liu, Shengju Qian, Xin Wang, Tao Hu, Ke Cao, Keyu Yan, and Jie Zhang. Id-animator: Zero-shot identity-preserving human video generation. *arXiv preprint arXiv:2404.15275*, 2024. 6, 7

[15] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 7

[16] Zhewei Huang, Tianyuan Zhang, Wen Heng, Boxin Shi, and Shuchang Zhou. RIFE: real-time intermediate flow estimation for video frame interpolation. *CoRR*, abs/2011.06294, 2020. 3

[17] Zhitong Huang, Nanxuan Zhao, and Jing Liao. Unicolor: A unified framework for multi-modal colorization with transformer. *ACM Transactions on Graphics (TOG)*, 41(6):1–16, 2022. 1

[18] Zhitong Huang, Mohan Zhang, and Jing Liao. Lvcd: reference-based lineart video colorization with diffusion models. *arXiv preprint arXiv:2409.12960*, 2024. 2, 3, 4, 6, 7

[19] Siddhant Jain, Daniel Watson, Eric Tabellion, Aleksander Holynski, Ben Poole, and Janne Kontkanen. Video interpolation with diffusion models. *CoRR*, abs/2404.01203, 2024. 3

[20] Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik G. Learned-Miller, and Jan Kautz. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018. 3

[21] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Co-tracker: It is better to track together. *arXiv preprint arXiv:2307.07635*, 2023. 4, 5

[22] Hyunsu Kim, Ho Young Jhoo, Eunhyeok Park, and Sungjoo Yoo. Tag2pix: Line art colorization using text tag with secat and changing loss. In *Int. Conf. Comput. Vis.*, pages 9056–9065, 2019. 2

[23] Hyun-Su Kim, HoYoung Jhoo, Eunhyeok Park, and Sungjoo Yoo. Tag2pix: Line art colorization using text tag with secat and changing loss. *Cornell University - arXiv,Cornell University - arXiv*, 2019. 2

[24] Changlin Li, Guangyang Wu, Yanan Sun, Xin Tao, Chi-Keung Tang, and Yu-Wing Tai. H-VFI: hierarchical frame interpolation for videos with large motions. *CoRR*, abs/2211.11309, 2022. 3

[25] Zekun Li, Zhengyang Geng, Zhao Kang, Wenyu Chen, and Yibo Yang. Eliminating gradient conflict in reference-based line-art colorization. In *European Conference on Computer Vision*, pages 579–596. Springer, 2022. 2

[26] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. Lightglue: Local feature matching at light speed. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17627–17638, 2023. 4, 5

[27] Yifan Liu, Zengchang Qin, Tao Wan, and Zhenbo Luo. Auto-painter: Cartoon image generation from sketch by using conditional wasserstein generative adversarial networks. *Neurocomputing*, 311:78–87, 2018. 2

[28] Zhiheng Liu, Ka Leong Cheng, Xi Chen, Jie Xiao, Hao Ouyang, Kai Zhu, Yu Liu, Yujun Shen, Qifeng Chen, and Ping Luo. Manganinja: Line art colorization with precise reference following. *arXiv preprint arXiv:2501.08332*, 2025. 2

[29] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60:91–110, 2004. 4, 5

[30] Yue Ma, Hongyu Liu, Hongfa Wang, Heng Pan, Yingqing He, Junkun Yuan, Ailing Zeng, Chengfei Cai, Heung-Yeung Shum, Wei Liu, et al. Follow-your-emoji: Fine-controllable and expressive freestyle portrait animation. *arXiv preprint arXiv:2406.01900*, 2024. 3

[31] Akinobu Maejima, Hiroyuki Kubo, Takuya Funatomi, Tatsuo Yotsukura, Satoshi Nakamura, and Yasuhiro Mukaigawa. Graph matching based anime colorization with multiple references. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference, SIGGRAPH 2019, Los Angeles, CA, USA, July 28 - August 1, 2019, Posters*, pages 13:1–13:2, 2019. 2

[32] Zhenglin Pan, Yu Zhu, and Yuxuan Mu. Sakuga-42m dataset: Scaling up cartoon research. *arXiv preprint arXiv:2405.07425*, 2024. 4, 5

[33] Amal Dev Parakkat, Pooran Memari, and Marie-Paule Cani. Delaunay painting: Perceptual image colouring from raster contours with gaps. *Computer Graphics Forum*, page 166–181, 2022. 1

[34] Xuebin Qin, Zichen Zhang, Chenyang Huang, Masood Dehghan, Osmar R Zaiane, and Martin Jagersand. U2-net: Going deeper with nested u-structure for salient object detection. *Pattern recognition*, 106:107404, 2020. 5

[35] Yingge Qu, Tien-Tsin Wong, and Pheng-Ann Heng. Manga colorization. *ACM Transactions on Graphics*, 25(3): 1214–1220, 2006. 2

[36] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 2

[37] Patsorn Sangkloy, Jingwan Lu, Chen Fang, Fisher Yu, and James Hays. Scribbler: Controlling deep image synthesis with sketch and color. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5400–5409, 2017. 2

[38] Kazuhiro Sato, Yusuke Matsui, Toshihiko Yamasaki, and Kiyoharu Aizawa. Reference-based manga colorization by graph correspondence using quadratic programming. In *SIGGRAPH Asia 2014 Technical Briefs*, page 1–4, 2014. 2

[39] Min Shi, Jia-Qi Zhang, Shu-Yu Chen, Lin Gao, Yu-Kun Lai, and Fang-Lue Zhang. Deep line art video colorization with a few references. *arXiv preprint arXiv:2003.10685*, 2020. 3

[40] Li Siyao, Shiyu Zhao, Weijiang Yu, Wenxiu Sun, Dimitris Metaxas, Chen Change Loy, and Ziwei Liu. Deep animation video interpolation in the wild. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6587–6595, 2021. 2, 3

[41] Daniel Sýkora, John Dingliana, and Steven Collins. Lazybrush: Flexible painting tool for hand-drawn cartoons. *Computer Graphics Forum*, 28(2):599–608, 2009. 2

[42] Daniel Sýkora, John Dingliana, and Steven Collins. Lazybrush: Flexible painting tool for hand-drawn cartoons. *Computer Graphics Forum*, page 599–608, 2009. 1

[43] Luming Tang, Menglin Jia, Qianqian Wang, Cheng Perng Phoo, and Bharath Hariharan. Emergent correspondence from image diffusion. *Advances in Neural Information Processing Systems*, 36:1363–1389, 2023. 5

[44] Harrish Thasarathan, Kamyar Nazeri, and Mehran Ebrahimi. Automatic temporally coherent video colorization. In *2019 16th conference on computer and robot vision (CRV)*, pages 189–194. IEEE, 2019. 2, 3

[45] Thomas Unterthiner, Sjoerd Van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018. 7

[46] Ning Wang, Muyao Niu, Zhi Dou, Zhihui Wang, Zhiyong Wang, Zhaoyan Ming, Bin Liu, and Haojie Li. Coloring anime line art videos with transformation region enhancement network. *Pattern Recognition*, 141:109562, 2023. 3

[47] Wen Wang, Qiuyu Wang, Kecheng Zheng, Hao Ouyang, Zhekai Chen, Biao Gong, Hao Chen, Yujun Shen, and Chunhua Shen. Framer: Interactive video interpolation. *arXiv preprint https://arxiv.org/abs/2410.18978*, 2024. 3

[48] Weijia Wu, Zhuang Li, Yuchao Gu, Rui Zhao, Yefei He, David Junhao Zhang, Mike Zheng Shou, Yan Li, Tingting Gao, and Di Zhang. Draganything: Motion control for anything using entity representation. In *European Conference on Computer Vision*, pages 331–348. Springer, 2025. 5

[49] Jinbo Xing, Hanyuan Liu, Menghan Xia, Yong Zhang, Xintao Wang, Ying Shan, and Tien-Tsin Wong. Tooncrafter: Generative cartoon interpolation. *arXiv preprint arXiv:2405.17933*, 2024. 3, 6, 7

[50] Jinbo Xing, Menghan Xia, Yong Zhang, Haoxin Chen, Wangbo Yu, Hanyuan Liu, Gongye Liu, Xintao Wang, Ying Shan, and Tien-Tsin Wong. Dynamicrafter: Animating open-domain images with video diffusion priors. In *European Conference on Computer Vision*, pages 399–417. Springer, 2024. 3

[51] Dingkun Yan, Liang Yuan, Erwin Wu, Yuma Nishioka, Issei Fujishiro, and Suguru Saito. Colorizediffusion: Adjustable sketch colorization with reference image and text. *arXiv preprint arXiv:2401.01456*, 2024. 1

[52] Jie Yang, Ailing Zeng, Ruimao Zhang, and Lei Zhang. X-pose: Detecting any keypoints. In *European Conference on Computer Vision*, pages 249–268. Springer, 2025. 5

[53] Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023. 6, 7

[54] Yifeng Yu, Jiangbo Qian, Chong Wang, Yihong Dong, and Baisong Liu. Animation line art colorization based on the optical flow method. *Computer Animation and Virtual Worlds*, 35(1):e2229, 2024. 2, 3

[55] Lvmin Zhang, Chengze Li, Tien-Tsin Wong, Yi Ji, and Chunping Liu. Two-stage sketch colorization. *ACM Transactions on Graphics*, page 1–14, 2018. 2

[56] Lvmin Zhang, Chengze Li, Tien-Tsin Wong, Yi Ji, and Chunping Liu. Two-stage sketch colorization. *ACM Trans. Graph.*, 37(6):1–14, 2018. 2

[57] Lvmin Zhang, Chengze Li, Edgar Simo-Serra, Yi Ji, Tien-Tsin Wong, and Chunping Liu. User-guided line art flat filling with split filling mechanism. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9889–9898, 2021. 2

[58] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Int. Conf. Comput. Vis.*, pages 3836–3847, 2023. 2

[59] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023. 4, 7

[60] Qian Zhang, Wang Bo, Wen Wang, Hai Li, and Liu Hui. Line art correlation matching feature transfer network for automatic animation colorization. *Cornell University - arXiv,Cornell University - arXiv*, 2020. 3

[61] Qian Zhang, Bo Wang, Wei Wen, Hai Li, and Junhui Liu. Line art correlation matching feature transfer network for automatic animation colorization. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3872–3881, 2021. 2

[62] Xingran Zhou, Bo Zhang, Ting Zhang, Pan Zhang, Jianmin Bao, Dong Chen, Zhongfei Zhang, and Fang Wen. Cocosnet v2: Full-resolution correspondence learning for image translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11465–11475, 2021. 2

[63] Tianyi Zhu, Wei Shang, Dongwei Ren, and Wangmeng Zuo. Thin-plate spline-based interpolation for animation line inbetweening. *arXiv preprint arXiv:2408.09131*, 2024. 2, 3

[64] Changqing Zou, Haoran Mo, Chengying Gao, Ruofei Du, and Hongbo Fu. Language-based colorization of scene sketches. *ACM Transactions on Graphics*, page 1–16, 2019. 2

[65] Chengyi Zou, Shuai Wan, Marc Gorriz Blanch, Luka Murn, Marta Mrak, Juil Sock, Fei Yang, and Luis Herranz. Lightweight deep exemplar colorization via semantic attention-guided laplacian pyramid. *IEEE Transactions on Visualization and Computer Graphics*, 2024. 1