

MangaNinja: Line Art Colorization with Precise Reference Following

Supplementary Material

The supplementary materials are structured as follows:

- We present and analyze additional possible solutions for reference-based line art colorization.
- A user study is conducted to further evaluate the superiority of our method, and a visual example is included to illustrate the benchmark we constructed for easier understanding.
- Extensive ablation studies are conducted, including experiments on the image feature extractor and the progressive patch shuffle training strategy.
- We provide more visual results of MangaNinja.

Finally, we sincerely invite you to review the **MP4** files in our supplementary materials, which contain visualizations of the relevant results.

Contents

A Analysis of More Possible Solutions	1
A.1. Analysis of line art video colorization methods	1
A.2. Analysis of RefOnly	2
A.3. Analysis of other styles	2
B User Study and Benchmark	2
B.1. User study	2
B.2. Visual illustration of our benchmark	2
C More Ablation Studies	3
C.1. Failure cases	3
C.2. Visual ablation of patch shuffle	3
C.3. Reference feature extractor	3
C.4. Progressive patch shuffle	3
D More Results	3

A. Analysis of More Possible Solutions

A.1. Analysis of line art video colorization methods

Recent advancements [1, 2] are being made in video line art colorization. ToonCrafter [2] is a video interpolation model that allows users to input line art as a control condition for colorization. LVCD [1] is another video line art colorization method, enabling users to colorize a sequence based on an initial video frame and subsequent line art frames. We explore the potential of using such video methods for image-based line art colorization. We find that video line art colorization models like these work well only for continuous and small variations in line art.

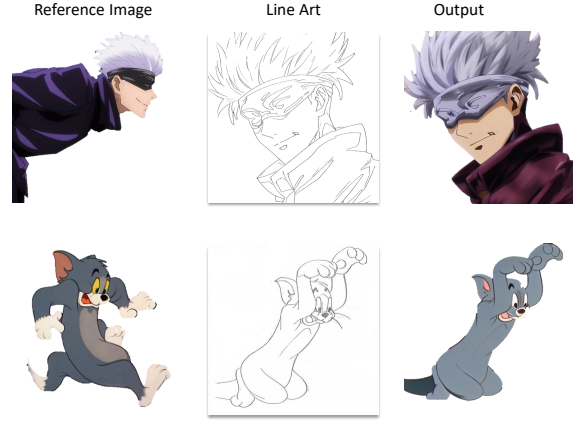


Figure S1. **Performance of video colorization methods on non-Continuous line art.** We select LVCD for experiments because the line art conditioned generation code for ToonCrafter is not available.

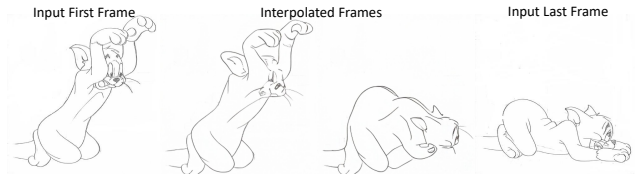


Figure S2. **Line art interpolation results.**

As shown in Fig. S1, when provided with a significantly different reference image and line art, these models fail to produce effective colorization. This limitation requires users to upload continuous line art sequences between the reference and target frames, which is impractical for single-image colorization.

One possible solution is to generate intermediate line art via interpolation to pass reference information to the target frame. However, as illustrated in Fig. S2, the current interpolation capabilities of video models have limitations. When there are large changes between the start and end frames, the generated interpolations lack consistency. We also observe a color leakage issue with LVCD. As shown in Fig. S3, when given non-binarized sketches, LVCD produces colorized results resembling the ground truth, even when the reference image is empty. This occurs due to the presence of low-value background regions that are invisible to the human eye (while the line art is represented as white with value 1, the background is nearly black, close to 0.) The network learns to map these subtle values



Figure S3. **Visualization of GT leakage.** The ‘original output’ represents the results from the paper. However, we find that when the reference image is empty, the model still generates outputs partially resembling the ground truth, as shown in ‘No Reference Output.’ Moreover, setting low-value background regions in grayscale sketches to zero severely degrades the colorization quality.

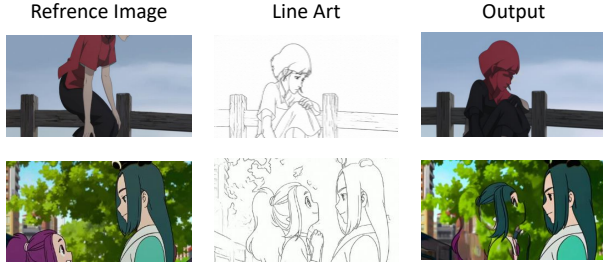


Figure S4. **Visualization of RefOnly.**

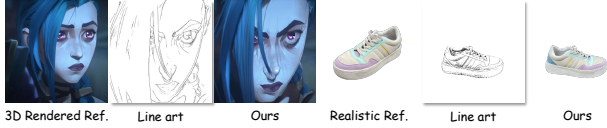


Figure S5. **Other reference art styles.**

to colors, causing ground truth leakage. Setting these low-value regions to zero eliminates the leakage, but it significantly degrades the colorization quality.

In contrast, MangaNinja does not suffer from these issues. During training, we set low-value background areas in grayscale sketches to zero to avoid ground truth leakage. By incorporating effective training strategies and explicitly injecting correspondence information, MangaNinja achieves precise matching, enabling accurate colorization even when there are large variations between the reference image and the line art.

A.2. Analysis of RefOnly

Another potential solution is to combine RefOnly with ControlNet [3]. As shown in Fig. S4, similar to LVCD, a simple RefNet primarily maps the colors of the reference image onto the line art based on approximate spatial distribution, lacking precise matching capabilities.



Figure S6. **Other line art styles.**

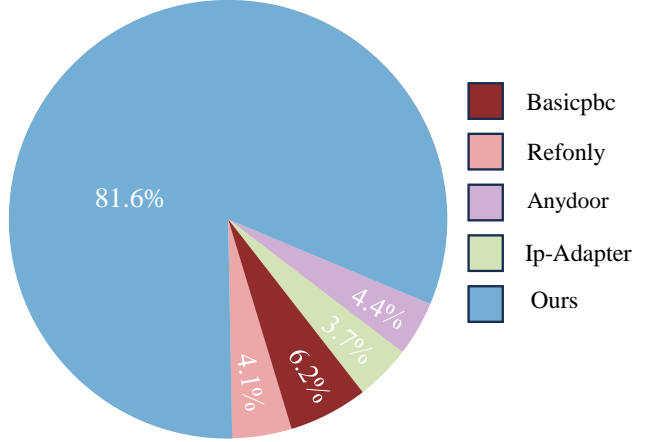


Figure S7. **User study results.**

A.3. Analysis of other styles

MangaNinja’s patch-based learning enables robust correspondence matching, supporting generalization to unseen styles. While trained on anime data for task alignment, as shown in Fig. S5, our method achieves strong results on realistic/3D line art without retraining. Our method uses line art extracted by a standard extractor (common in practice and close to hand-drawn styles) during training. As shown in Fig. S6, MangaNinja achieves robust results even on raw hand-drawn inputs without retraining.

B. User Study and Benchmark

B.1. User study

To further compare all methods, we also conduct a user study. Specifically, we select 40 pairs of reference images and line art for automatic colorization. We invite twenty participants, and each is asked to choose the method that produces the highest quality and most accurate color matching. As shown in Fig. S7, our method demonstrates a clear advantage over the others.

B.2. Visual illustration of our benchmark

To provide a clearer explanation of the benchmark we constructed, we have selected an example for visualization. As shown in Fig. S8, the background of the selected image is removed to avoid affecting metric calculations. Notably, in addition to common metrics, we provide matching points

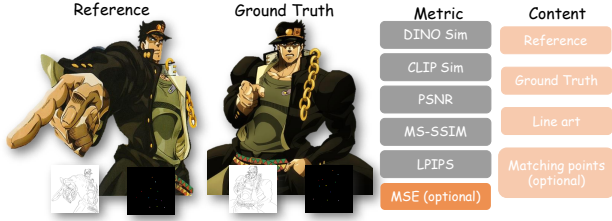


Figure S8. Illustration of constructed benchmark.

to calculate pixel-level MSE, which is used to evaluate the fine details of the colorization.

C. More Ablation Studies

C.1. Failure cases

Our performance is constrained by the input sketch’s quality. As shown in Fig. S9, it may degrade if the line art is incomplete or distorted.

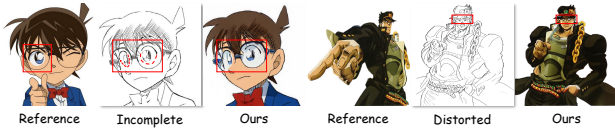


Figure S9. Failure cases.

C.2. Visual ablation of patch shuffle

We further visualize ablation results of progressive patch shuffle. As shown in Fig. S10, when patch shuffle is not used, the model is more likely to remember simple structural information, resulting in artifacts of the reference image. Random shuffle instead of progressive shuffle will prevent the model from learning accurate matching capabilities.



Figure S10. Visual ablation of patch shuffle.

C.3. Reference feature extractor

MangaNinja employs a dual-branch U-Net to extract image features from the reference image and the target image, respectively. To validate the effectiveness of the Reference U-Net structure in learning matching capabilities, we replace the reference U-Net with DINO and CLIP image encoders, injecting 16×16 patch tokens within the cross-attention layers, while keeping all other training settings

identical. The experiments are conducted using the same dataset and experimental settings. All training strategies are applied in the same way. As shown in Fig. S11, compared with the Reference U-Net, using CLIP or DINO as the encoder results in weaker handling of fine details.

C.4. Progressive patch shuffle

The purpose of our patch shuffle strategy is to disrupt the structural information in the reference image, preventing the model from learning a simple offset to perform colorization. Instead, we want the model to develop a finer-level matching ability. Thus, how to effectively disrupt the structure becomes a key question. We find that using overly fine-grained shuffles during training (e.g., dividing the patches into 32×32 segments) makes it difficult for the model to converge. On the other hand, using a coarse shuffle (e.g., dividing into 2×2 patches) fails to adequately break down the structural information. Therefore, we adopt a coarse-to-fine learning scheme by progressively increasing the number of randomly shuffled patches. Specifically, we multiply the number of shuffled patches by four every 40k steps.

As shown in Tab. S1, we apply patch shuffle to the base model without additional training strategies or point guidance. With an increasing number of shuffled patches, the model’s performance improves consistently until the difference between 32×32 and 64×64 becomes negligible. Therefore, we ultimately set the shuffled patch size to 32×32 .

D. More Results

To further demonstrate the precise matching capability of MangaNinja, we provide additional visual results. As shown in Fig. S12, we colorize all characters from the manga *One Piece*. As shown in Fig. S13, we use the same reference with different line art images in row one and the same line art with different references in row two. The results show that our method exhibits strong robustness.

References

- [1] Zhitong Huang, Mohan Zhang, and Jing Liao. Lvcd: Reference-based lineart video colorization with diffusion models. *arXiv preprint arXiv:2409.12960*, 2024. 1
- [2] 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. 1
- [3] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *International Conference on Computer Vision*, pages 3836–3847, 2023. 2



Figure S11. Comparison of different image feature extractors.

Table S1. **Ablation of patch shuffle.** Note that each model in the table inherits from the previous one, with an increased number of shuffled patches used for training.

Number of shuffled patches	DINO \uparrow	CLIP \uparrow	PSNR \uparrow	MS-SSIM \uparrow	LPIPS \downarrow
2×2	63.91	84.75	18.02	0.912	0.27
4×4	64.42	85.23	18.44	0.924	0.25
8×8	65.13	85.87	18.77	0.935	0.25
16×16	66.69	86.49	19.24	0.943	0.24
32×32	67.12	86.93	<u>19.72</u>	<u>0.952</u>	0.23
64×64	67.09	<u>86.86</u>	19.88	0.954	0.23



Figure S12. Colorization results for *One Piece* characters.



Figure S13. **More visualization results.** One reference to colorize multiple line art images in row one; multiple references to colorize the same sketch in row two.