

Appendix

A. Further Implementation Details

We set the regularization weight to be $5e3$ for the $\mathcal{L}_{\text{legibility}}$ and $1e3, 1e4$ for the λ_1, λ_2 in $\mathcal{L}_{\text{structure}}$. We observe that $\mathcal{L}_{\text{legibility}}$ often plays a dominant role. When perceptual regularization is employed from the start, the canonical shape typically retains the original letter’s form, preventing any semantic deformations. Hence, following Iluz et al. [2], we gradually increase the weight of $\mathcal{L}_{\text{legibility}}$ to make its effects after semantic deformation has taken place.

We use the text-to-video-ms-1.7b model in ModelScope [3, 7] for the diffusion backbone. We apply augmentations, including random crop and random perspective, to all video frames. We intend to make our code available to release all the details and support further research.

B. Optimization Efficiency

A visually plausible result often converges within 200 iterations (~ 8 mins), as shown in Fig. 1. The subsequent optimization further refines motion quality and enriches detail. One promising direction to improve efficiency involves leveraging our generated animation data to fine-tune Text-to-Video (T2V) models, as in FlipSketch [1].

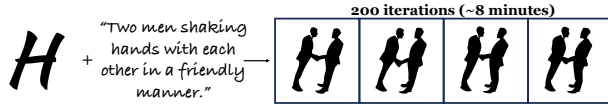


Figure 1. A visually plausible Dynamic Typography with only 200 iterations optimization (about 8 minutes).

C. Frequency-based Encoding and Annealing

NeRF [4] has highlighted that a heuristic application of sinusoidal functions to input coordinates, known as “positional encoding”, enables the coordinate-based MLPs to capture higher frequency content, as denoted by:

$$\gamma(p) = \left(\sin(2^0 \pi p), \cos(2^0 \pi p), \dots, \sin(2^{L-1} \pi p), \cos(2^{L-1} \pi p) \right), \quad (1)$$

where p refers to the point coordinates.

We found that this property also applies to our MLPs with coordinates of control points as input. This allows the MLPs in the canonical and deformation fields to more effectively represent high-frequency information, corresponding to the detailed geometric features. Additionally, when using coordinate-based MLPs to model motion, a significant challenge is capturing both minute and large motions. Following Nerfies [6], we employ a coarse-to-fine strategy that initially targets low-frequency (large-scale) motion and progressively refines high-frequency (localized)

motions. Specifically, we use the following formula to apply weights to each frequency band j in the positional encoding of the MLPs within the deformation field.

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

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

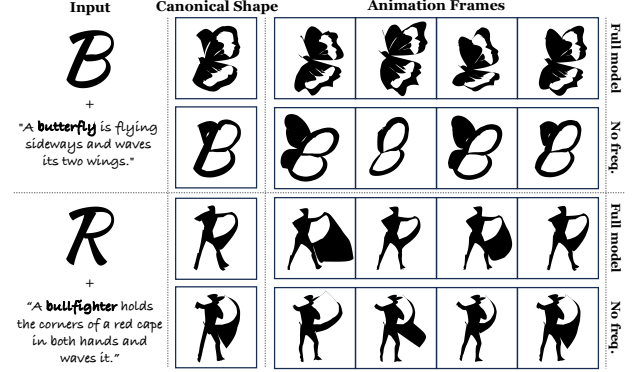


Figure 2. Visual comparison with and without frequency encoding and annealing. The geometry and motion quality degrade when removing annealed frequency-based encoding.

Fig. 2 shows the visual ablation comparisons with and without the frequency encoding and annealing. When removing annealed frequency-based encoding, the geometry and motion quality of the generated animation suffer. To be specific, the butterfly animation in Fig. 2 (row 2) exhibits a lack of geometry details, and the bullfighter animation in Fig. 2 (row 4) shows unreasonable motion, leading to the degradation of the animation quality.

D. Generalizability over Different Fonts, Prompts, and Languages

Our method generalizes well in generating Dynamic Typography over different fonts, text prompts, and languages. When the letter to be animated takes different fonts, the generated animation faithfully preserves the unique stylistic elements of each font, including serif details, stroke weights, and overall character proportions, as illustrated in Fig. 3. This style preservation ensures that the brand identity or design intention embedded in the font choice remains intact throughout the animation sequence.

In Fig. 4, by adjusting the input prompts, our method exhibits remarkable flexibility in generating diverse animation effects for the same letter. This prompt-driven control allows designers to explore various creative possibilities,

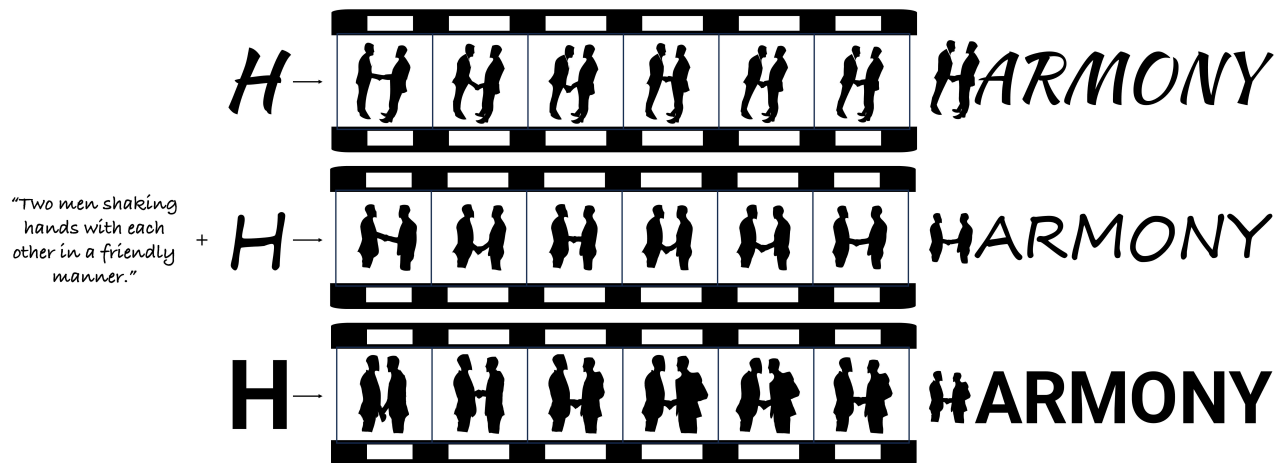


Figure 3. Dynamic Typography over different fonts for the same animation sample. The corresponding fonts in the first, second, and third rows are KaushanScript-Regular, Segoe Print, and Roboto-Bold, respectively. The animated letter "H" preserves the unique style of each font while faithfully depicting the prompts, allowing it to be seamlessly integrated into the word "HARMONY" under different fonts.

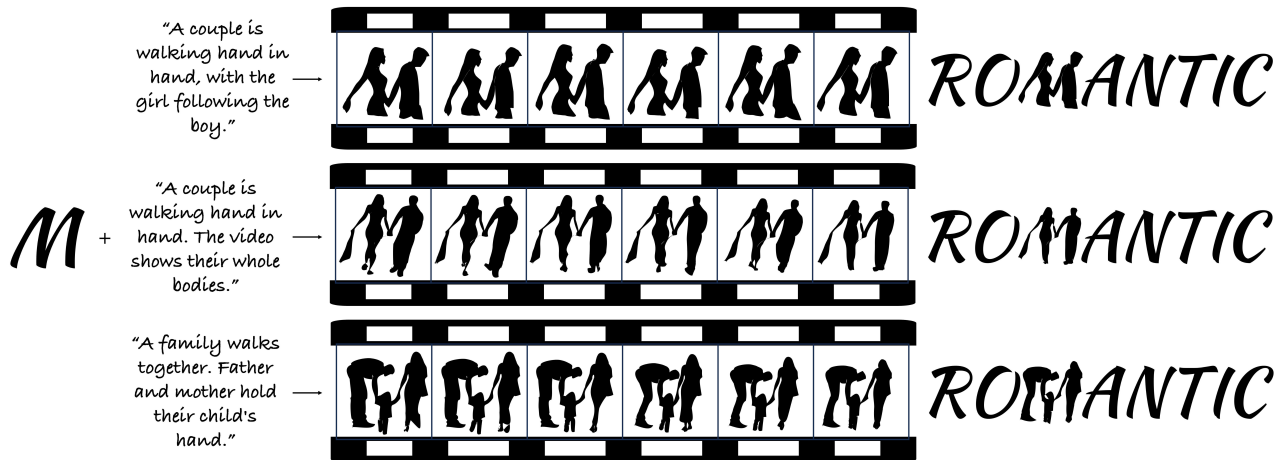


Figure 4. Dynamic Typography over different prompts for the same letter "M" to be animated. Our approach displays different visual effects based on three different prompts, offering a completely different reading experience for the same word.

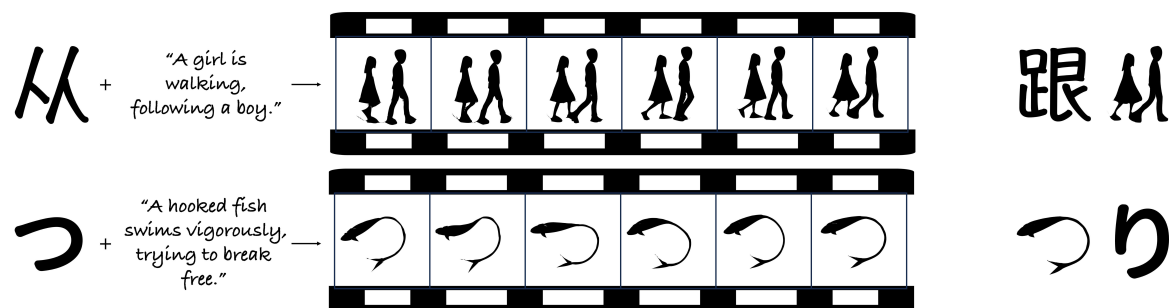


Figure 5. Dynamic Typography over different languages. In the first row, we animate the Chinese character "从" in the word "跟从", meaning "following" in English. In the second row, we animate the hiragana character 「つ」 from the Japanese word 「つり」, meaning "fishing" in English. This demonstrates the potential of our proposed methodology to generate Dynamic Typography over different languages.

effectively turning text descriptions into distinct visual narratives. The coherent animations demonstrate our method’s ability to interpret creative intentions while maintaining visual quality.

Furthermore, Fig. 5 showcases our method’s language adaptability through Dynamic Typography samples in Chinese and Japanese. These examples not only demonstrate successful handling of complex writing systems with different structural characteristics from the Latin alphabet but also indicate the potential for broader applications across various writing systems and cultural contexts.

E. Role of Homography Term

The homography term, as referenced in Equation (3) of the main text, plays a crucial role in enabling more expressive and dynamic movements within the animation. This term models global transformations, which means it can apply changes like rotation, scaling, and perspective shifts to the entire animated shape across different frames. This capability is essential for capturing complex motions that go beyond simple, localized deformations of the shape itself.

As illustrated in the figure Fig. 6, the animation sequence generated without the homography term (top row) shows a relatively rigid fish-like shape with limited movement.

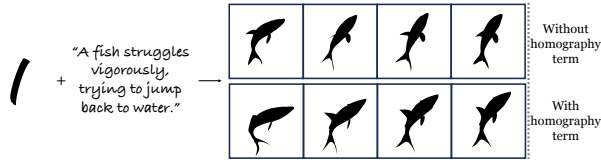


Figure 6. Visual comparison demonstrating the role of the homography term. The top row, created without the homography term, shows a rigid motion. The bottom row, created with the homography term, allows for global transformations that result in a more dynamic and expressive animation, better capturing the essence of the prompt.

F. Prompt Effect Analysis

Fig. 7 further shows results from a simple and a complex prompt, both producing visually appealing animation, suggesting that extensive manual prompt engineering is not always necessary.

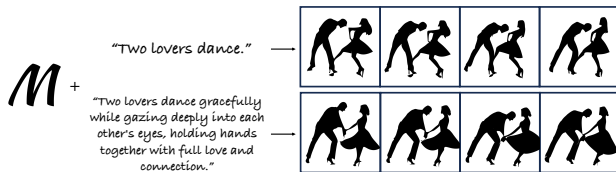


Figure 7. Dynamic Typography over a simple and a complex prompt for the same letter “M”, both producing vivid results.

G. Fine-Grained User Control

Beyond manipulating individual key control points, users can freely customize the rendering styles, such as the color and line style of the animated letter, as illustrated in the main text. Users also have the option to directly provide the canonical shape for animation. The following Fig. 8 shows a text animation sample of the letter “Y” in “BUNNY” with the user-provided canonical shape.



Figure 8. Dynamic Typography of “Y” in “BUNNY”. User directly provide a canonical shape of a bunny instead of the letter.

Users might desire varying degrees of legibility throughout an animation. Our current framework allows users to control this by adjusting the **displacement weight** for each frame, which influences the displacement magnitude with respect to the canonical shape. Users have the flexibility to customize this weighting scheme. As an example in App. G, users can set the displacement weight to reach its minimum value at 12-th frame, thus imposing the strongest legibility constraint during the middle of the animation.

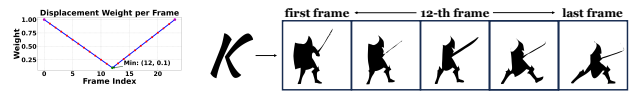


Figure 9. A user-defined displacement weight curve (left) controls the legibility of an animation. By setting the minimum weight at the 12-th frame, the animated character’s silhouette (right) is constrained to most closely resemble the canonical “K” shape in the middle of the sequence.

H. Failure Case

We observe that, in some samples, the semantic meaning and corresponding motion specified by the user-provided text prompt significantly diverged from the original shape of the letter. In such cases, the model struggles to simultaneously maintain the shape of the letter and convey the vivid semantic information of the text prompt. As a result, the letter either undergoes minimal change in shape, retaining its original form, or it completely loses its original shape, compromising legibility.

For example, in Fig. 11, when we incorporate the legibility regularization, the shape of the letter “R” remains unchanged, maintaining its original form while performing the “launch” action. Conversely, when we reduce the weight of the legibility regularization, it transforms completely into the shape of a rocket, losing the characteristic contours of the letter “R”, thus sacrificing the legibility.

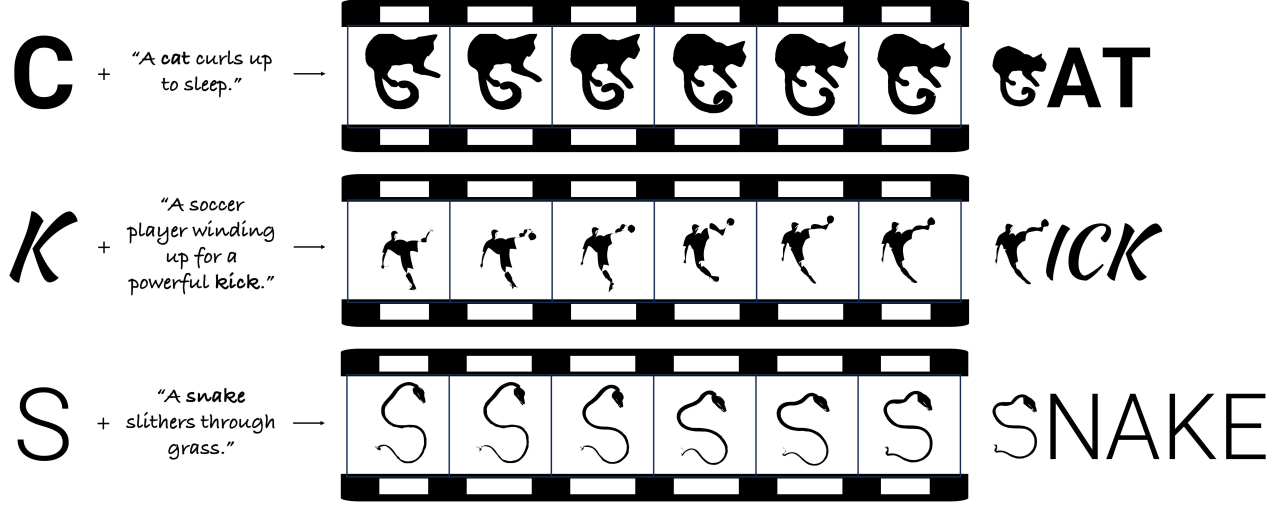


Figure 10. Some results generated by our method based on pairs of prompts and letters that are designed with the assistance of GPT-4V.

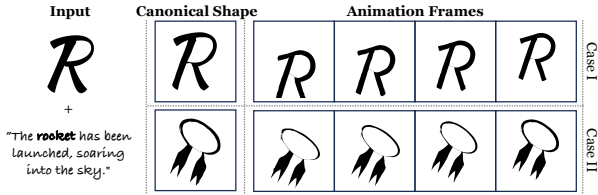


Figure 11. Failure case illustration. The first row is generated with the default weight for legibility and structure preservation loss, which suffers from minimal semantic deformation. In the second row, we reduce the weight of these regularization losses, which compromises the legibility.

I. GPT-4V as Dynamic Typography Designer

As illustrated in Fig. 11, if the user-specified text prompt deviates too much from the chosen letter’s shape, it can hinder the creation of vivid animations. We can utilize the powerful visual and semantic understanding capabilities of Vision Language Models (VLMs) to assist users in selecting appropriate letters and prompts.

In the experiment, we provide current state-of-the-art VLM, GPT-4V [5], with a snapshot of an animation generated, along with the corresponding chosen word, letter, and text prompt as an example to facilitate in-context learning by GPT-4V. Subsequently, we request GPT-4V to design text animations by following the paradigm exemplified in the previous experiment. We require GPT-4V to generate outputs including the word, the selected letter, and the text prompt, explicitly demanding that it considers the similarity between the letter’s original shape and its shape after deformation.

We list some samples designed by GPT-4V as follows:

Word: CAT
Chosen Letter: C
Text Prompt: “A cat curls up to sleep.”
Animation Idea: The “C” naturally curls tighter into a circular shape, resembling a cat curling up.

Word: KICK
Chosen Letter: K
Text Prompt: “A soccer player kicks a ball.”
Animation Idea: The angled legs of the “K” mimic the motion of kicking, with one leg drawing back and then striking forward.

Word: SNAKE
Chosen Letter: S
Text Prompt: “A snake slithers through grass.”
Animation Idea: The natural curve of “S” undulates slightly, resembling the slithering movement of a snake.

The generated animations designed by GPT-4V are shown in Fig. 10. We found that GPT-4V has the potential to design proper pairs of prompts and letters that carefully consider the natural shapes of the letters and how they can effectively transform into the desired actions or characteristics with minimal deviation, ensuring the animations are feasible and visually coherent.

This success with GPT-4V suggests promising future directions in leveraging more advanced multimodal language models (MLLMs) or AI agents for animation design.

References

- [1] Hmrishav Bandyopadhyay and Yi-Zhe Song. Flipsketch: Flipping static drawings to text-guided sketch animations. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 28394–28404, 2025. [1](#)
- [2] Shir Iluz, Yael Vinker, Amir Hertz, Daniel Berio, Daniel Cohen-Or, and Ariel Shamir. Word-as-image for semantic typography. *ACM Trans. Graph.*, 42(4), 2023. [1](#)
- [3] Zhengxiong Luo, Dayou Chen, Yingya Zhang, Yan Huang, Liang Wang, Yujun Shen, Deli Zhao, Jingren Zhou, and Tieniu Tan. Videofusion: Decomposed diffusion models for high-quality video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. [1](#)
- [4] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. [1](#)
- [5] OpenAI. Gpt-4 technical report, 2024. [4](#)
- [6] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5865–5874, 2021. [1](#)
- [7] Jiuniu Wang, Hangjie Yuan, Dayou Chen, Yingya Zhang, Xiang Wang, and Shiwei Zhang. Modelscope text-to-video technical report. *arXiv preprint arXiv:2308.06571*, 2023. [1](#)