Prompt Enhancer LLM for Video Generation

Daria Kryvosheieva
Massachusetts Institute of Technology
Cambridge, MA 02139
`daria_k@mit.edu`

Abstract

Video generation models, such as OpenAI's Sora 2, can create remarkably beautiful videos. However, achieving an aesthetically pleasing video that shows exactly what the user wants often requires an elaborate prompt, whereas videos generated from simple prompts tend to contain artifacts or differ from what the user originally imagined. We train a language model to translate simple prompts into detailed, professional-grade prompts, eliminating the need for prompt engineering and enabling users to generate high-quality Sora videos at low effort cost. [1]

1 Introduction

Since OpenAI developed the first version of Sora [2] in early 2024, video generation models have advanced rapidly, and are currently capable of generating videos of remarkable aesthetic quality. Given these capabilities, the impressiveness of a generated video now depends largely on the user: creative plots, striking visual details, and lively motion effects must be specified in the prompt, whereas short, simple prompts typically yield correspondingly simple videos. Moreover, videos generated from more elaborate prompts tend to exhibit fewer artifacts. This means we are still far away from a "one-click blockbuster" ideal: users who aim for showcase-ready videos need to invest substantial time and effort into prompt engineering.

For example, a popular (3.2K likes) video[2] by one of Sora's most subscribed users (`@keigo_matsumaru`; 93K subscribers) was generated via the following prompt:

```
# Cosmic Arena Reverie ## Subject / Scene Settings - Audience: {locale="JP"}; Narrative tone: awe,
liminal-dream - Subject type: environment - Key features: navyviolet nebula-like smoke; stadium bowl
with LED ring; reflective wet field; ember-orange core glow; starry particles; Scale: arena-scale;
Motion: slow drift/eddies, light ripples - Location: futuristic stadium interior; Time: night;
Weather/Light: humid mist, volumetric haze lit by LEDs/cosmic glow - Key elements (FG/MG/BG): FG
mirror-wet ground & low fog / MG pitch lines & drifting smoke sheets / BG tiered seats + LED ring +
hanging rigs - Lighting: soft vol light from overhead nebula sweeping LR; LED ring as rim; neg fill
from dark stands; subtle bounce off wet ground; mild gobo from truss; haze medium; vol present -
Grade: Palette deep navy/electric blue/violet/ember orange; soft-contrast curve, lifted blacks; gentle
halation; subtle vignette; grain fine; mild chromatic aberration; clean arc flares from LEDs - Visual
taste: cinematic sci-fi reverie; crisp highlights, dreamy tails; Background/Location: enclosed bowl,
hints of glass roof - Camera: WSMS; centered symmetry with stadium curves; parallax from ring;
occluders: smoke veils; ONE move gimbal slow push with slight arc - Lens/Focus: 2435mm feel; creamy
bokeh from LED ring; gentle rack ground reflections  overhead cloud - Coverage: master push + inserts
(LED ring, water ripples); match-on-action for smoke swirl; keep clockwise screen direction; eyelines
N/A - Persist: same LED ring layout & palette; wet reflective pitch; clockwise parallax maintained ##
Dialogue (concise; speaker labels consistent) - [02s] Announcer (whisper): "Welcome to the infinite
arena." - [24s] Whisper: "Tonight, the sky takes the field." ## Audio (BGM & SFX) - BGM: airy pads +
sub pulses + shimmer keys (80 BPM; long intro swell, gentle end fade) - SFX: distant crowd hush, LED
buzz, soft whooshes, electrical crackle, water drip ticks, airy sparkles - Cues: 0.05s twin
```

[1] **GitHub:** https://github.com/dariakryvosheieva/video-prompt-enhancer
HuggingFace: https://huggingface.co/dariakryvosheieva/video-prompt-enhancer

[2] https://sora.chatgpt.com/p/s_68e5421db0c48191a97bdd357b87203b

Final project for MIT 6.7920: Reinforcement Learning.

Besides being very long and using a structured template, this prompt contains filmmaking jargon unfamiliar to many ordinary users (*parallax*, *24-35mm lens*, *BGM & SFX*, *80 BPM*).

We train a large language model (LLM) to translate simple video generation prompts written by ordinary users into detailed, professional-grade prompts like the one above. Our training pipeline has two stages:

1. **Next-token prediction** on pairs of simple and corresponding detailed prompts;
2. **Online RL**:
    - 2.1. The model accepts a simple prompt and generates a detailed prompt;
    - 2.2. A Sora video is generated based on the detailed prompt;
    - 2.3. The video and its alignment with the simple prompt are scored using VisionReward [9];
    - 2.4. The model is updated via PPO [7].

We validate that videos generated from the model's output prompts align with human preferences by posting them on a public Sora account[3].

2 Stage 1

First, we adapt the model to the target domain by performing large-scale, cheap SFT training.

2.1 Data

The training dataset for Stage 1 consists of 1,200 synthetic pairs of simple prompts and corresponding ground-truth detailed prompts. To maximize the diversity of topics, visual styles, and prompt styles, the data points were generated from diverse LLMs (GPT-5.1 [6], Claude Sonnet/Haiku 4.5 [1], Gemini 2.5 Pro/Flash [3], DeepSeek-V3.2 [4], Qwen3 VL [8]). Where applicable, both reasoning and non-reasoning modes were used. All LLMs were given few-shot examples based on real detailed prompts from popular Sora users `@keigo_matsumaru`, `@hakoniwa`, and `@kejia`.

Initially, we generated 1,000 data points, but upon inspection, we observed that the resulting videos are almost exclusively realistic. Therefore, we dedicated 50 more data points to each of the four selected non-realistic styles: anime, 3D animation, 2D cartoon, and pixel art.

2.2 Model

We initialize the model from the pre-trained and instruction-tuned backbone `Qwen2.5-14B-Instruct` [10]. Due to the relatively large size of the model and the fact that the target task is not very different from standard instruction-following, we use LoRA [5] (on all attention and MLP weight matrices) through HuggingFace's PEFT library. We use the following LoRA hyperparameters: $r = 16$, $\alpha = 32$, $\mathrm{dropout} = 0.05$.

2.3 Training procedure

At each step, we sample a batch of prompt pairs. For each pair in the batch, we feed the simple prompt (along with an instruction string; see Figure 1) into the model and compute the cross-entropy loss against the ground-truth detailed prompt. We use a batch size of 8 and a learning rate of $2 \cdot 10^{-4}$.

```
f"Convert the following video generation prompt into a professional-grade
prompt that will produce a high quality, aesthetic, and impressive video. If
the original prompt includes a style specification (such as 'anime', 'pixel',
or 'cartoon'), keep it in the converted prompt. Output only the converted
prompt.\n\nInput:\n{simple_prompt.strip()}\n\nOutput:\n"
```

Figure 1: The instruction provided to the model.

After this training stage, the model's output prompts already result in better videos compared to simple prompts, as evidenced by examples[4].

[3] https://sora.chatgpt.com/profile/daria-k

[4] https://drive.google.com/drive/folders/1TNZNtibuatasdkwiXHeTF9ezlynuboe6?usp=sharing

3 Stage 2

We refine the model's capabilities using a small number of online RL updates, which, despite being expensive, provide a strong learning signal.

3.1 Data

For this stage, the training dataset consists of simple prompts only. We generated 100 synthetic simple prompts following the approach in Section 2.1.

3.2 Model

We resume training the LoRA adapters (for the same backbone `Qwen2.5-14B-Instruct`) from the checkpoint saved after Stage 1.

For the purpose of PPO training, we add a value head—a linear layer that will be trained to predict the expected reward of the detailed prompt—on top of the LM head.

3.3 Training procedure

At each step, we sample a batch of simple prompts. For each simple prompt in the batch, we feed it into the LLM—along with the same instruction string as in Figure 1—and obtain the output detailed prompt. We then use Sora 2 API to generate a video based on the detailed prompt. We score the video using the specialized reward model VisionReward [9] based on its aestheticity, absence of artifacts, and alignment with the original simple prompt. Finally, we use the earned reward to perform a PPO update on the LoRA weights.

We conduct PPO training using HuggingFace's TRL library. PPO updates weights by maximizing a clipped policy objective

$$\mathcal{L} = \mathbb{E}_t[\min(r_t A_t, \text{clip}(r_t, 1-\epsilon, 1+\epsilon)A_t)],$$

where t is a token index in the detailed prompt, r_t is the reward for token t, and A_t is the advantage. For the final token, the reward consists of the VisionReward score and the KL penalty; for intermediate tokens, this is KL penalty only. Advantages are computed via *generalized advantage estimation* (GAE) using the value head V_θ's predictions as a learned baseline:

$$A_t = \sum_l (\gamma\lambda)^l \delta_{t+l} \quad \text{where} \quad \gamma = 1, \;\; \lambda = 0.95, \;\; \delta_t = r_t + \gamma V_\theta(s_{t+1}) - V_\theta(s_t)$$

We use a batch size of 2, resulting in a total of 50 PPO steps. We sample detailed prompts with a temperature of 0.7 and a `top_p` value of 0.95. We generate 4-second videos at a 720x1280 resolution. For PPO, we set the learning rate to $5 \cdot 10^{-6}$, the target KL to 0.1, and ϵ to 0.2.

See top-3 highest- and lowest-reward videos produced during this training stage[5].

3.4 Evaluation

Figure 2 shows mean VisionReward score (average over the two rewards in a batch) as a function of batch index.

The reward dynamics looks noisy, and the slope of the best fit line is slightly negative. The noisy dynamics is expected given the small number of steps (50), the randomness of detailed prompt and video generation (both the prompt enhancer LLM and Sora use sampling), and the diversity of topics and styles.

It is worth noting that true objective we aim to optimize is the alignment of the generated videos with human preferences. VisionReward only acts as a proxy whose alignment with the true objective itself should be validated. We observe that VisionReward is biased toward assigning high rewards to videos with a beautiful but static visual track (no motion), while assigning low rewards to non-realistic styles (even when the original prompt explicitly states that the video should be in a cartoonish or 3D-animated style).

[5] https://drive.google.com/drive/folders/163ED2SuYSLbzEk3_r8SQ32PP1spgol9e?usp=sharing

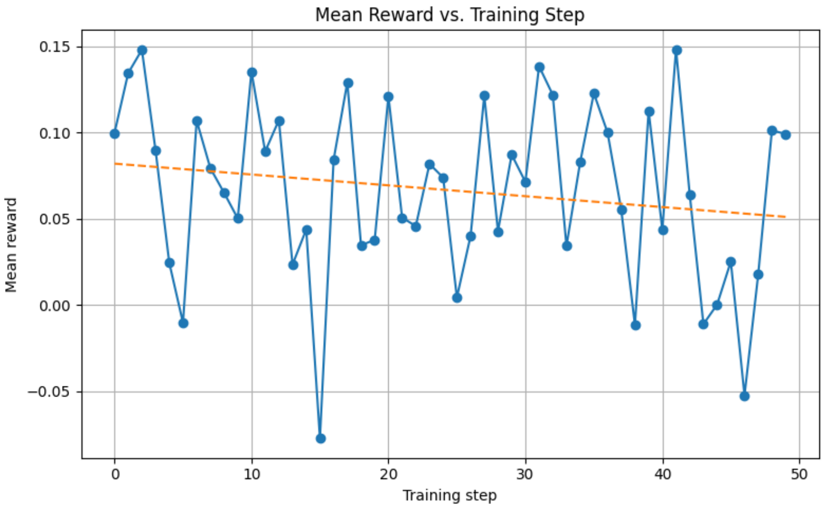

Figure 2: Mean reward as a function of time step.

To evaluate alignment with human preferences directly, we post videos generated from the LLM's rewritten prompts on a public Sora account (`@daria_k`). Each of the 3 posted videos earned an average of 2.33 likes, confirming that the videos are appealing to real users.

4 Training Details

Both training stages were conducted on one H200 GPU on MIT's Engaging cluster. Stage 1 took approximately 20 minutes, and Stage 2 took approximately 8 hours.

Limitations

We acknowledge that the quality of generated videos is hard to evaluate: mean VisionReward score and number of user likes reflect two different dimensions of quality. Therefore, we are not entirely sure whether performance improved after Stage 1 compared to the backbone and after Stage 2 compared to Stage 1. In industry, alignment with human preferences is evaluated via large-scale human validation trials, where thousands of participants rate the model's outputs.

If we had more time and money, we would use more data for both Stages 1 and 2. We would also conduct a survey in which participants would rank videos generated from (i) simple prompts, (ii) detailed prompts produced by the base Qwen model, (iii), detailed prompts after Stage 1, and (iv) detailed prompts after Stage 2, to see clearly how each factor influences the human preference objective.

References

[1] Anthropic. System card: Claude sonnet 4.5, 2025.

[2] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024.

[3] Google DeepMind. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities, 2025.

[4] DeepSeek-AI. Deepseek-v3.2: Pushing the frontier of open large language models, 2025.

[5] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

[6] OpenAI. Gpt-5.1 instant and gpt-5.1 thinking system card addendum, 2025.

[7] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017.

[8] Qwen Team. Qwen3-vl technical report, 2025.

[9] Jiazheng Xu, Yu Huang, Jiale Cheng, Yuanming Yang, Jiajun Xu, Yuan Wang, Wenbo Duan, Shen Yang, Qunlin Jin, Shurun Li, Jiayan Teng, Zhuoyi Yang, Wendi Zheng, Xiao Liu, Dan Zhang, Ming Ding, Xiaohan Zhang, Xiaotao Gu, Shiyu Huang, Minlie Huang, Jie Tang, and Yuxiao Dong. Visionreward: Fine-grained multi-dimensional human preference learning for image and video generation, 2025.

[10] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2024.