A Full Related Work

A.1 Video Diffusion Model

Video generation has become a rapidly growing focus in generative AI. Modern approaches typically use a VAE to compress videos into a latent space, where a diffusion model—conditioned on text, images, or both—learns to generate content. Early studies, such as Make-A-Video (Singer et al., 2022), PYoCo (Ge et al., 2023) and Tune-A-Video (Wu et al., 2023b), adapted text-to-image models with additional temporal layers to enable video generation. Works like, MagicVideo (Zhou et al., 2022), SVD (Blattmann et al., 2023a) and Latent Video Diffusion (Blattmann et al., 2023b) played pioneering roles in scaling latent diffusion approaches. However, the limited compression rate of VAEs has hindered their ability to generalize to long video sequences. A major breakthrough came with Sora (Brooks et al., 2024), which introduced a temporal VAE to compress temporal dimensions alongside spatial ones, while adopting a transformer-based backbone (Peebles & Xie, 2023) at scale. Recent efforts have pushed this framework further. For instance, Wan 2.2 (Wang et al., 2025a) incorporated a sparse MoE architecture that routes different diffusion steps to specialized experts, while VEO3 (DeepMind, 2025) extended the paradigm by integrating audio, achieving state-of-the-art performance. The success of MovieGen (Polyak et al., 2024), Seaweed (Seawead et al., 2025), Goku (Chen et al., 2025b), and Waiver (Zhang et al., 2025d) further demonstrates the potential of video generation and its broad impact on practical applications. These developments underscore video generation as one of the most dynamic and competitive frontiers in generative AI community.

A.2 Auto-regressive Diffusion Model

Auto-regressive generation dominates the text domain, while diffusion models have become the standard for visual generation. Recent research explores how to combine these paradigms to duplicate the long-term planning capacity of large language models in vision generation. A straightforward solution (Zhou et al., 2024) is to jointly train an autoregressive (AR) model for text and a diffusion model for vision, but this leaves the visual side reliant solely on diffusion without benefiting from AR modeling. Inspired by block diffusion (Arriola et al., 2025), several works (Li et al., 2024b; Hu et al., 2024; Deng et al., 2024; Ren et al., 2025; 2024) explore AR–diffusion hybrids: MAR (Li et al., 2024b) disentangles the two, letting AR predict conditions and diffusion reconstruct tokens; ACDiT (Hu et al., 2024) integrates them via block-wise diffusion with autoregression across blocks, while CausalDiffusion (Deng et al., 2024) extends this to token-level autoregression. Extending these ideas to video is natural since frames form temporal chunks. FAR (Gu et al., 2025) generates each frame autoregressively; Dini (Liu et al., 2024a) employs an AR planner to provide frame-level conditions, with diffusion recovering pixels for tasks such as video interpolation, video extension, and image-to-video generation. Beyond this, MAGI (Teng et al., 2025) and Skyreel (Chen et al., 2025a) remove the dual-model design, training under the strategy of diffusion forcing (Chen et al., 2024a), where later frames are assigned higher noise levels, thereby enabling infinite autoregressive inter-chunk prediction and high-quality inner-chunk diffusion generation. More recently, self-forcing (Huang et al., 2025) highlights a gap between training (real data diffused with noise) and inference (model-generated conditions), and proposes rollout-based training to align the two, leading to more robust long-term prediction.

A.3 Efficient Attention for Multimodal Generation.

Diffusion Transformers (DiT) have emerged as the mainstream architecture for visual content generation. Representative models include PixArt-α (Chen et al., 2023b), Stable Diffusion 3 (SD3) (Esser et al., 2024), and Flux (Labs, 2024), the latter demonstrating the potential of scaling DiT to 12B parameters for high-resolution image synthesis.. To address the computational challenges of vanilla attention ($O(n^2)$)), various methods have replaced it with linear-complexity mechanisms. For instance, DiG (Zhu et al., 2024) uses gated linear attention, PixArt-Σ (Chen et al., 2024b) designs key-value token compression, while LinFusion (Liu et al., 2024b) involves Mamba-based structure and SANA (Xie et al., 2025a) employs ReLU linear attention approaches to reduce computational overhead. With the rise of video generation, the computational demands of standard quadratic attention have become a major bottleneck. To address the high computational cost of 3D video attention, many existing works employ factorized spatial and temporal attention to reduce complexity (Chen et al., 2023a; Ho et al., 2022; Wang et al., 2025c; Singer et al., 2022). Other methods reduce at-

tention complexity by selectively skipping certain token interactions (Xi et al., 2025; Yang et al., 2025; Li* et al., 2025; Zhang et al., 2025a;b;c). Simultaneously, other models, such as Mamba-based architectures (Wang et al., 2025b; Gao et al., 2024), have explored state-space models and linear-complexity designs for efficient video generation. However, these methods either retain some quadratic complexity due to global self-attention layers or are limited to local attention. In contrast, our model maintains a constant-memory KV cache with global attention mechanism, enabling the generation of high-quality, minute-length videos.

B More Implementation Details

B.1 Pipeline Configuration

As detailed in Table 6, our SANA-Video-2B model supersedes the original SANA (Xie et al., 2025a) architecture, including the diffusion transformer and a small decoder-only text encoder, to best utilize the pre-trained text-to-image model's weights. However, we introduce several key modifications to support video generation. We increase the FFN dimension from 5600 to 6720 and the head dimension from 32 to 112 to accommodate 3D RoPE, and we add a temporal convolution in the Mix-FFN module to enhance motion performance. To effectively capture latent features from both images and videos, our approach uses different VAEs based on resolution. For 480P videos, we leverage a Wan2.1-VAE (Wang et al., 2025a) to prioritize reconstruction quality with a lower compression rate (F8T4C16). In contrast, for high-resolution 720P videos, we fine-tune the DCAE (Chen et al., 2024c) into a more aggressive deep compression autoencoder, DCAE-V (F32T4C32), to facilitate more efficient training and inference. For conditional feature extraction, we follow SANA by using a small decoder-only LLM for efficient text processing. For our training strategy, we also employ multi-aspect augmentation to enable arbitrary aspect ratio generation and facilitate image-video joint training, allowing the model to generate both images and videos from a single architecture. The AdamW optimizer (Loshchilov & Hutter, 2017) is utilized with a weight decay of 0.03 and a constant learning rate of 5e-5. We use Accelerate FSDP (acc, 2022) for efficient sharded data parallel training. Our final model is trained on 64 H100 GPUs for approximately 12 days.

Table 6: Architecture details of the proposed SANA-Video.

Model	Width	Depth	FFN	#Heads	#Param (M)
SANA-Video-2B	2240	20	6720	20	2056

B.2 Block Linear Attention during Inference

We follows Self-Forcing (Huang et al., 2025) for autoregressive inference, with the KV cache update based on our design (Alg. 1). Specifically, we first initialize KV cache as empty and start to denoise the first block. After it is fully denoised, the attention state $\sum_0^0 S$, cumulative sum of keys $\sum_0^0 \phi(K)^T$ and conv cache f will be stored. For the remaining blocks (*e.g.*, n-th block), they will use the existing KV cache to denoise the latent until clean and then update the cumulative attention state $\sum_0^n S$ and cumulative sum of keys $\sum_0^n \phi(K)^T$. Also, conv cache f will be replaced with the new cache. Such update leverages the global while keeping the memory constant and small, making the long video generation efficient and effective.

C More Results

Please refer to our **anonymous link** (https://sana-video.pages.dev/), for the qualitative comparison and our generation results.

C.1 VAE Comparison

In Sec. 3.4, we analyze the differences and performance of various video VAEs. To select the VAE that best suits our small diffusion model, we conducted a generalization experiment. We hypothesize that a VAE with better reconstruction ability under perturbation will be a better fit, as the diffusion

Algorithm 1 Block Linear Diffusion Inference with Linear KV Cache

Require: KV cache
Require: Denoise timesteps $\{t_1, \ldots, t_T\}$, noise scheduler Ψ
Require: Number of blocks M
Require: Block-wise diffusion model G_θ (G_θ^{KV} returns cumulative sum of state $\sum S$, cumulative sum of key $\sum \phi(K)^T$ and conv cache f)

Initialize model output $\mathbf{X}_\theta \leftarrow []$
Initialize KV cache $\mathbf{KV} \leftarrow [\text{None}, \text{None}, \text{None}]$
for $i = 1, \ldots, M$ **do**
  Initialize $x_{t_T}^i \sim \mathcal{N}(0, I)$
  for $j = T, \ldots, 1$ **do**
    Set $\hat{x}_0^i \leftarrow G_\theta(x_{t_j}^i; t_j, \mathbf{KV})$
    if $j = 1$ **then**
      $\mathbf{X}_\theta$. `append`$(\hat{x}_0^i)$
      Update Cache $\mathbf{KV} \leftarrow G_\theta^{\text{KV}}(\hat{x}_0^i; 0, \mathbf{KV})$
    else
      Sample $\epsilon \sim \mathcal{N}(0, I)$
      Set $x_{t_{j-1}}^i \leftarrow \Psi(\hat{x}_0^i, \epsilon, t_{j-1})$
    end if
  end for
end for
return $\mathbf{X}_\theta$

model's output during inference may be slightly different from the clean latent distribution seen during VAE training. Specifically, we add Gaussian noise to the encoded latent before decoding it, setting $\boldsymbol{x}_t' = \boldsymbol{x}_t + \epsilon \boldsymbol{z}$, where $\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$. As the results in Table 7 show, our DCAE-V performs much more robustly under different noise levels. This demonstrates its superior reconstruction generalization, making it the ideal choice for our small diffusion model.

Table 7: Performance comparison of different VAE models on 1000 samples from Panda-70M with different noise perturbation levels.

Model	latent shape	psnr↑	ssim↑	lpips↓
Wan2.1VAE ($\epsilon = 0$)	16, T/4, H/8, W/8	34.41	0.95	**0.01**
Wan2.2VAE ($\epsilon = 0$)	48, T/4, H/16, W/16	**35.61**	**0.96**	**0.01**
DCAE-V ($\epsilon = 0$)	32, T/4, H/32, W/32	33.25	0.94	0.03
Wan2.1VAE ($\epsilon = 0.1$)	16, T/4, H/8, W/8	28.61	0.89	0.06
Wan2.2VAE ($\epsilon = 0.1$)	48, T/4, H/16, W/16	30.12	0.92	**0.04**
DCAE-V ($\epsilon = 0.1$)	32, T/4, H/32, W/32	**31.91**	**0.93**	**0.04**
Wan2.1VAE ($\epsilon = 0.2$)	16, T/4, H/8, W/8	24.25	0.78	0.16
Wan2.2VAE ($\epsilon = 0.2$)	48, T/4, H/16, W/16	25.94	0.84	0.10
DCAE-V ($\epsilon = 0.2$)	32, T/4, H/32, W/32	**29.34**	**0.90**	**0.05**

C.2 Qualitative Comparison

Text-to-Video Generation. We compare the text-to-video generation results with current state-of-the-art small diffusion models Wan2.1-1.3B (Wang et al., 2025a) and Wan2.2-5B (Wang et al., 2025a). As shown in Fig. 7, SANA-Video has comparable semantic understanding, great motion control, and high aesthetic quality.

Image-to-Video Generation. We compare the image-to-video generation results with small diffusion models LTX-Video (HaCohen et al., 2024) (2B) and SkyReelv2-I2V (Chen et al., 2025a) (1.3B). As shown in Fig. 8, SANA-Video has the best semantic understanding ability ("camera remains steady" instruction in the first case) as well as the best motion control ("slow-motion effect" instruction in the second case) and moderate motion magnitude (first case).

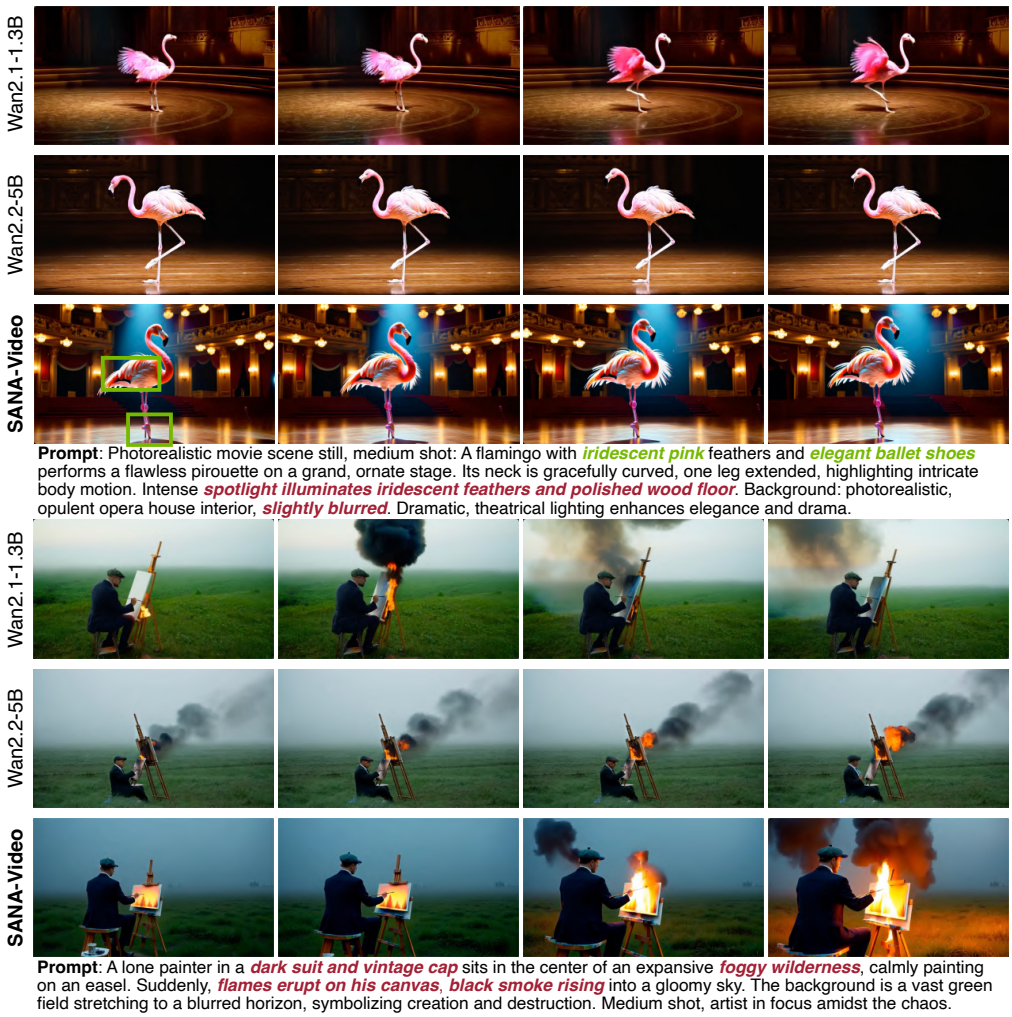


Figure 7: **Qualitative comparison among T2V methods.** SANA-Video has comparable motion control and video-text semantic alignment with state-of-the-art small diffusion models.

C.3 More I2V Results

Our SANA-Video is a unified framework that can perform T2I, T2V and I2V with a single model. We visualize the I2V generation results in Fig. 9. The first column is the reference image and the remaining columns are the generated video. Our SANA-Video can generate semantic consistent and temporal smooth videos based on the first frame.

C.4 Influence of Motion Score

As mentioned in our data pipeline (Sec. D), we use the average optical flow value to represent the motion magitude, which is called motion score in our paper. The motion score is added to the text prompt to better control the motion. In Fig. 10, we compare the impact of motion score in the I2V task, which is more clear with the same reference image. By increasing the motion, SANA-Video can generate videos with larger but still consistent motion.

C.5 Long Video Visualization

In Fig. 11, we provide an example of our 30-second long video generation. SANA-Video is able to generate motion consistent and semantically aligned long videos.

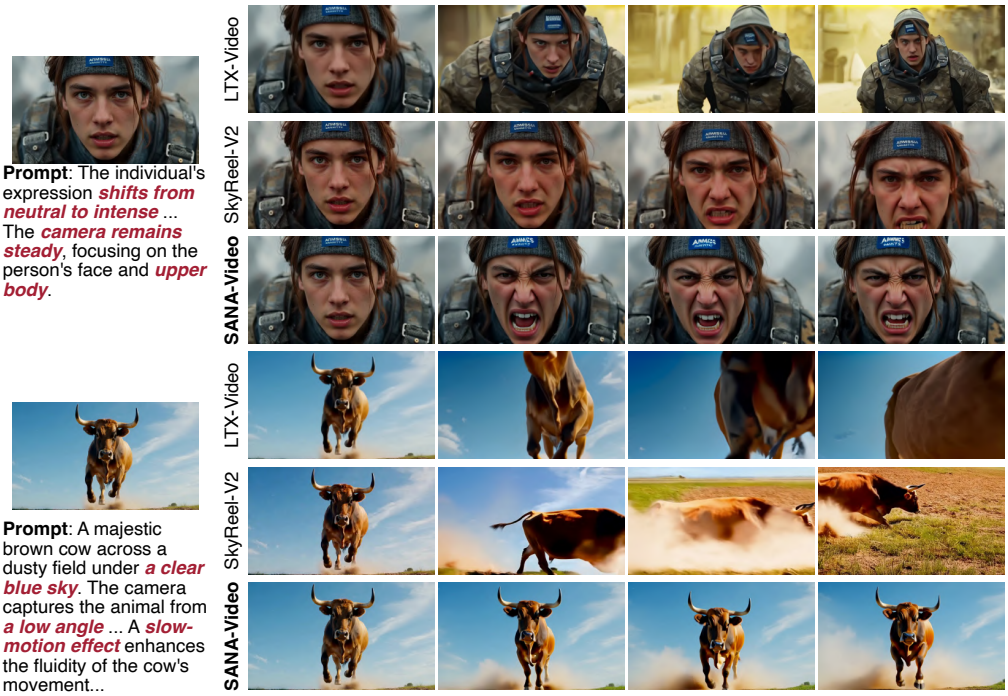


Figure 8: **Qualitative comparison among I2V methods.** SANA-Video has better motion control and video-text semantic alignment.

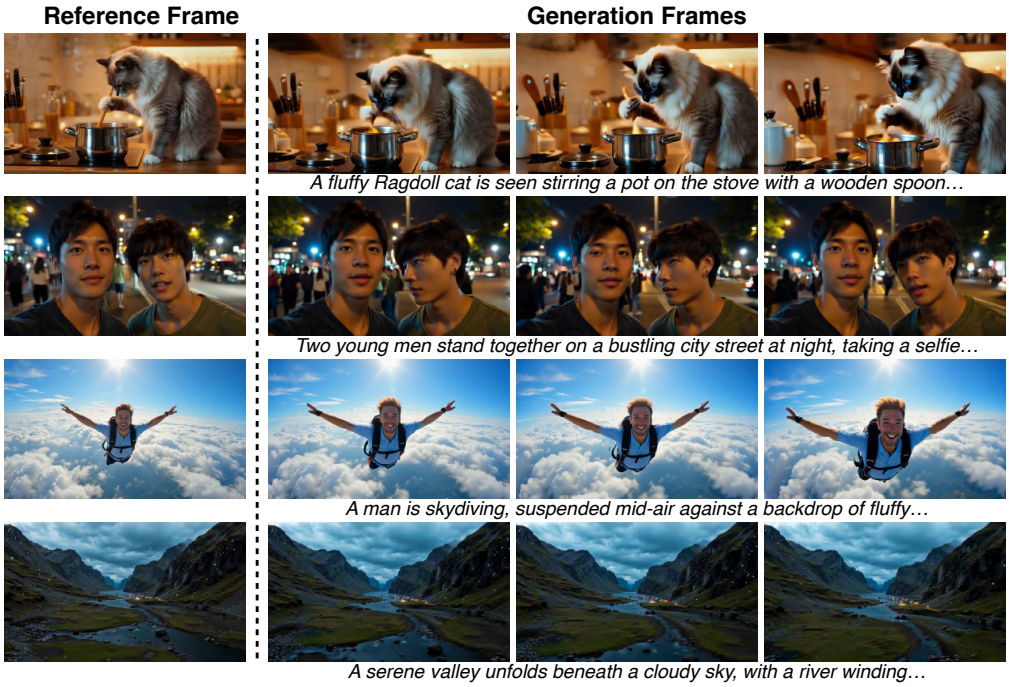
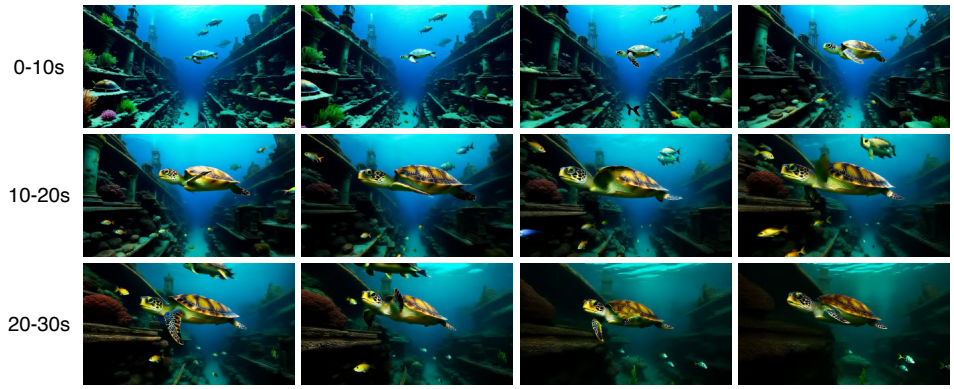


Figure 9: **Visualization of image-to-video generation.** SANA-Video can keep consistent with the first frame while generating realistic motion.

D Data Processing Pipeline

To build our training dataset, we collect public real and synthetic data and implement a multi-stage filtering paradigm. First, we use PySceneDetect (Castellano) and FFmpeg (Developers, 2025) to cut

Figure 10: **The impact of motion score on I2V task.** Higher motion score can lead to larger motion.



A deep-sea submersible explores a trench teeming with glowing fish, ancient ruins, and strange sea creatures, following a sea turtle gliding gracefully through the reef. Low angle, slow tracking shot.

Figure 11: **Long video visualization of SANA-Video.**

raw videos into single-scene short clips. For each video clip, we analyze its aesthetic and motion quality, as well as providing detailed captions. Specifically, the motion quality is measured by Unimatch (Xu et al., 2023) (optical flow) and VMAF (Peng et al., 2025) (pixel difference), and only clips with moderate and clear motion are kept. Furthermore, the average optical flow is used as a representation of motion magnitude, injecting into prompt for better motion controllability. Aesthetic quality is measured by a pre-trained video aesthetic model (DOVER (Wu et al., 2023a)) and key frame saturation obtained with OpenCV (Bradski, 2000), where low aesthetic score and over-saturated videos are removed. Finally, we collect approximately 5,000 human preferred high-quality videos based on stringent motion and aesthetic criteria. The SFT data is collected with diverse but balanced motion and style categories, which can further improve the overall performance. As shown in Fig. 12, the details of this data processing pipeline are discussed as follows.

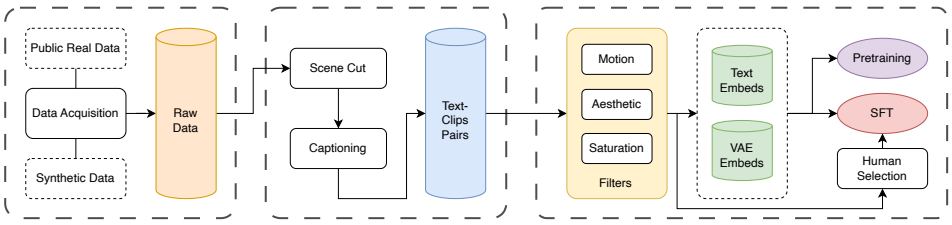


Figure 12: Data filtering paradigm of SANA-Video.

Scene Detection and Shot Cut. In the pre-training stage, we focus on generating 5-second short videos with 16 FPS on a specific scene. However, the raw videos are commonly long and contains more than one scene. Therefore, we cut the raw videos to small video shots with two steps: PySceneDet (Castellano) to split the scenes and FFmpeg (Developers, 2025) to split videos into short clips.

Motion Filtering. Our pre-training dataset comes from multiple sources, and each source of data differs not only in style but also in motion. Motion that is too fast or too slow degrades the motion performance of SANA-Video. Following Zheng et al. (2024), we apply Unimatch (Xu et al., 2023) and Vmaf to score the motion of each video. Unimatch can evaluate the optical flow of two given images of the same shape. We select frames from each video every 0.5 seconds, reshape them into

Figure 13: An overview of the captioning pipeline.

320x576, and calculate the average optical flow over all selected frames. Vmaf, on the other hand, simply computes the pixel difference of two images; we use FFmpeg (Developers, 2025) to compute the Vmaf over all consecutive frames and normalize them. Due to the variance of different video sources, we analyze the motion scale and set the appropriate motion range individually, ensuring our data has moderate and clear motion. During pre-training, we also append *Motion score:* {*unimatch value*} to the text prompt to help control the motion magnitude of the generated videos (Fig. 10).

Aesthetics Many Text-to-Image works have proven that high aesthetic data can improve the training efficiency of an image generation model (Chen et al., 2023b). We believe that this also applies to the video generation model. We use Dover (Wu et al., 2023a) to score each video for its aesthetics. Dover produces three different scores: aesthetic score, technical score, and overall score, among which we use the overall score as the filter metric.

Saturation We also observe that some of our data, especially synthetic data and real data converted from HDR to SDR, has unnatural color, appearing in high saturation. To prevent these data from damaging the output quality of SANA-Video, we use OpenCV (Bradski, 2000) to compute a saturation score of each video. We select frames from each video every 0.5 seconds, convert their color representation from RGB to HSV, where the "S" channel in HSV color representation stands for saturation. By averaging the "S" channel over all pixels and frames, we obtain the saturation score of a video. We keep only videos with a saturation score lower than a threshold set to a reasonable value for each data source.

Captioning (Wang et al., 2025a) shows that LLM rewritten prompts can produce more accurate and detailed prompts within the same distribution, and thus make the model easier to learn, and consequently enhance the model's performance. Moreover, for synthetic data with existing prompts, replacing their prompts with LLM rewritten ones helps reduce the misalignment between the original prompts and their synthetic output. Following (Wang et al., 2025a), we use Qwen-2.5-VL (Bai et al., 2025b) to caption our data as shown in Fig. 13.

SFT Data For our final stage of SFT training, we selected approximately 5,000 high-quality videos based on stringent criteria for motion and aesthetics. The motion requirement is fulfilled by the presence of either distinct object motion, camera motion, or both. Ideal object motion is characterized by a moderate magnitude and a clearly focused action that is free from occlusions. Similarly, any camera movement must be stable and smooth, without jittering, to maintain 3D consistency. The aesthetic criteria are equally comprehensive. Beyond technical qualities like balanced brightness and natural color, we prioritize videos with appealing overall content and layout, demonstrated by thoughtful composition and engaging subject matter. Following this filtering process, the videos were classified into four motion categories (human activities, animal activities, other objects, natural or urban scenes) and three aesthetic styles (realistic, cartoon, cinematic). This strategic sampling across diverse categories is crucial for ensuring both the model's high performance and the breadth of its capabilities. The influence of fine-tuning on the SFT data is illustrated in Fig. 14, where both the aesthetic details (the eyes in the first example), and the motion realism (the pipe of the second example) will be improved.

Figure 14: **Analysis of the influence of SFT.** Fine-tuning on the human preferred SFT data can improve the video details and adherence to the laws of physics.

E World Model

Figure 15: **Visualization of world model task generation.**

We fine-tune SANA-Video on several downstream tasks to demonstrate the potential of applying SANA-Video to world model related generation: embodied AI, autonomous driving and game generation.

World Model for Embodied AI. The first important downstream task for video generation is embodied AI, where SANA-Video can be used to generate simulation data for robot training. In this task, we leverage AgiBot (contributors, 2024) as the training data, which contains synchronized views of different camera views. The head-front view is adopted as the target videos and filtered with our data pipeline. The generation results are shown in the first row of Fig. 15.

World Model for Autonomous Driving. Video generation model is also a good simulator for autonomous vehicle scenarios, and SANA-Video can be used to generate diverse and realistic driving

scenes. In this task, we fine-tune SANA-Video on internal driving data, using the front camera with 30 FOV. The generation results are shown in the second row of Fig. 15.

World Model for Game Generation. We explore downstream game generation to create interactive video games. Specifically, we use VPT (Baker et al., 2022) as the training data, containing screen recording videos of players playing Minecraft. The raw videos are cut and processed following our data pipeline in Appendix D. In addition, we train a small classifier to identify low-quality data in the scenario. The generation results are shown in the third row of Fig. 15.

F LLM USAGE

Our use of large language models (LLMs) was limited to editorial assistance to improve the clarity and readability of this manuscript. Specifically, these tools were used to refine grammar and phrasing, enhance the logical flow between sections, and condense overly verbose passages for conciseness. Crucially, all original research ideas, experimental designs, and data analyses were conceived and executed by the authors; the LLM did not contribute to any scientific or methodological content.