FlashDecoder: Real-Time Latent-to-Pixel Streaming Decoder with Transformers

Supplementary Material

This supplementary material provides training specifications (Section A), inference protocols (Section B), additional visual results (Section C), and limitations and future directions (Section D).

A. Training Specifications

A.1. Dataset Details

Image Data. We utilize DataComp-small [5], comprising 12.8M image-text pairs. During preprocessing, we apply probabilistic augmentation that randomly selects among random cropping (40%), center cropping (30%), or resizing (30%) to the target resolution. Images smaller than the target resolution are filtered to prevent upsampling artifacts.

Video Data. Our video corpus combines Kinetics-600 [2] and an internal high-resolution collection of approximately 200K clips. From each video, we sample 17 consecutive frames at native frame rate. Preprocessing employs a two-stage spatial transformation: frames are first resized so that the shorter side matches the target resolution (480p, 720p, or 1080p depending on the training stage) while preserving the original aspect ratio, then cropped to the target resolution using either center crop (60%) or random crop (40%). Videos below the target resolution are filtered out. All resizing uses anti-aliased PIL bicubic interpolation [10, 17].

A.2. Multi-Stage Training Protocol

FlashDecoder follows a sequential three-stage training protocol, with each stage building upon the previous one. Training hyperparameters are summarized in Table A.

Stage 1: Low-Resolution Pre-training. This stage establishes fundamental reconstruction capabilities at reduced computational cost. We train on 224×224×17 video clips and 256×256 images with a 2:8 image-to-video sampling ratio to balance temporal coherence with spatial fidelity. Training proceeds for 200K iterations with batch size 16. The reconstruction objective combines L1 loss and perceptual loss [21] with weights of 1.0 and 0.1, respectively.

Stage 2: High-Resolution Training. We transition to higher resolutions to minimize the domain gap between training and inference. The model is trained on 480p clips (480×832×17), 720p clips (720×1280×17), 1080p clips (1080×1920×17), and 512×512 images. This diverse resolution mixture enables the model to handle varying spatial resolutions during inference. We reduce the learning rate by 10× and train for 100K iterations with batch size 8. Loss weights are adjusted to 1.0 for L1 and 0.25 for perceptual loss. The perceptual loss is computed on random 224×224 crops to keep memory manageable at high resolutions.

Stage 3: Adversarial Post-training. To enhance fine-grained details, we introduce adversarial training using the same data configuration as Stage 2 but excluding 1080p clips due to the additional memory overhead of the discriminator. This stage enables the decoder to synthesize sharper high-frequency textures that reconstruction losses alone cannot capture. We extend VQGAN's 2D PatchGAN discriminator [4, 9] to 3D for spatiotemporal processing, and train it with non-saturating logistic loss [7] and R1 regularization [14]. Both the perceptual and adversarial losses are computed on random 224×224 crops of the decoded output.

B. Inference Protocols

For fair comparison, we re-evaluate all baseline models using their official repository implementations and released checkpoints under identical settings on a single NVIDIA H100 GPU (80GB).

B.1. Throughput Measurement

To ensure fair comparison, we evaluate each model in its officially supported inference mode. Wan2.2-TAEHV [1], Wan2.2 [20], and FlashDecoder natively support streaming and are evaluated accordingly. Other VAEs (HunyuanVideo, AToken, MAGI-1) process entire clips in batch mode using their official implementations, as forcing them into a streaming setup would require chunking and blending that degrades their reconstruction quality. Throughput is measured in frames per second (FPS), calculated as total decoded frames divided by total decoding time.

C. More Visual Results

Figures A–C show additional 720p reconstruction results from Wan2.2-TAEHV [1], AToken [13], Wan2.2 [20], and FlashDecoder-XL-Opt. Wan2.2-TAEHV struggles with fine details; AToken produces smoother but blurrier outputs. Both Wan2.2 and FlashDecoder-XL-Opt produce sharp results, with Wan2.2 showing slightly finer details in some cases. FlashDecoder-XL-Opt achieves over 9× higher throughput (151.0 vs. 16.1 FPS).

D. Limitations and Future Directions

Decoder-only design. FlashDecoder replaces only the decoder while keeping the pretrained convolutional encoder fixed. The latent space therefore inherits the encoder's characteristics, which favor spatial locality by design; what properties a Transformer-based encoder–decoder

Table A. **Hyperparameters for FlashDecoder-XL training on the Wan2.2 latent space.** We report the training configurations for each stage. Stage 1 focuses on low-resolution pre-training, Stage 2 transitions to high-resolution training, and Stage 3 introduces adversarial post-training. For additional technical details, please refer to the original papers: LPIPS [21], R1 regularization [14], AdamW optimizer [12], and 3D PatchGAN [3, 9]. DDP denotes Distributed Data Parallel.

Hyperparameters	Stage 1	Stage 2	Stage 3
Model Architecture			
Latent channels (C')	48	48	48
Model dimension (D)	1536	1536	1536
# of Transformer blocks	20	20	20
# of Temporal refinement Transformer blocks	2	2	2
Attention heads (N)	24	24	24
KV groups (G)	3	3	3
MLP expansion	4.0	4.0	4.0
Temporal compression (r_{t})	4	4	4
Spatial compression (r_{s})	16	16	16
Window size (W_{frm})	2	2	2
Data Configuration			
Video resolution	224×224×17	480p / 720p / 1080p ×17	480p / 720p ×17
Image resolution	256×256	512×512	512×512
Sampling ratio	8:2 (video:image)	2:4:2:2 (480p:720p:1080p:image)	2:6:2 (480p:720p:image)
Loss Configuration			
L1 loss weight (λ_{L1})	1.0	1.0	1.0
Perceptual loss weight (λ_{LPIPS})	0.1	0.25	0.25
Adversarial loss type	-	-	Logistic
Adversarial loss weight (λ_{adv})	-	-	1e-4
R1 regularization weight	-	-	0.1024
R1 interval	-	-	16
Decoder & Decoder-related Optimization			
Optimizer	AdamW	AdamW	AdamW
Batch size	16	8	8
Learning rate	1e-4	1e-5	1e-5
AdamW β_1	0.9	0.9	0.9
AdamW β_2	0.999	0.999	0.999
Weight decay	0.01	0.01	0.01
EMA	-	-	0.9999
EMA warmup step	-	-	2000
Precision	bfloat16	bfloat16	bfloat16
Discriminator & Discriminator-related Optimization			
Architecture	-	-	3D PatchGAN
# of conv layers	-	-	5
Base channels	-	-	128
Learning rate	-	-	1e-5
AdamW β_1	-	-	0.0
AdamW β_2	-	-	0.9
Weight decay	-	-	0.01
Precision	bfloat16	bfloat16	bfloat16
Training specifications			
Training iterations	200K	100K	20K
Distributed training	DDP	DDP	DDP
GPU type	H100 80GB	H100 80GB	H100 80GB
# GPUs	8	8	8

pair would learn remains unexplored. Designing a streaming Transformer encoder to pair with FlashDecoder and training the full VAE from scratch is a clear next step. Such an encoder–decoder pair would also eliminate the resolution gap between low-resolution VAE training (256×256) and high-resolution diffusion training (720p, 1080p), since our streaming architecture scales to high resolutions. This is particularly relevant for recent end-to-end frameworks such as Unified Latents [8], which jointly train the VAE and diffusion model. Existing convolutional video VAEs are difficult to jointly train with a diffusion model at 480p or 720p due to their high memory consumption. FlashDecoder's low memory footprint makes such joint training feasible. Training the VAE and diffusion model at the same resolution used during inference would ensure that the latent space has good diffusibility at that resolution, avoiding potential mismatches caused by training at a lower resolution.

rFVD gap. FlashDecoder-XL falls short of Wan2.2 [20] and HunyuanVideo [11] in rFVD [6, 18], despite comparable PSNR and LPIPS. We trained on a single 8-GPU node, whereas these production decoders likely used significantly more compute and data. Scaling up model capacity and adversarial training duration is expected to close this gap.

Integration with Representation Autoencoders. Recent work on Representation Autoencoders (RAE [22]) pairs frozen pretrained encoders (e.g., DINOv2 [15], SigLIP [16]) with Transformer decoders for image generation. Extending this paradigm to video with a streaming-capable decoder like FlashDecoder is a promising direction.

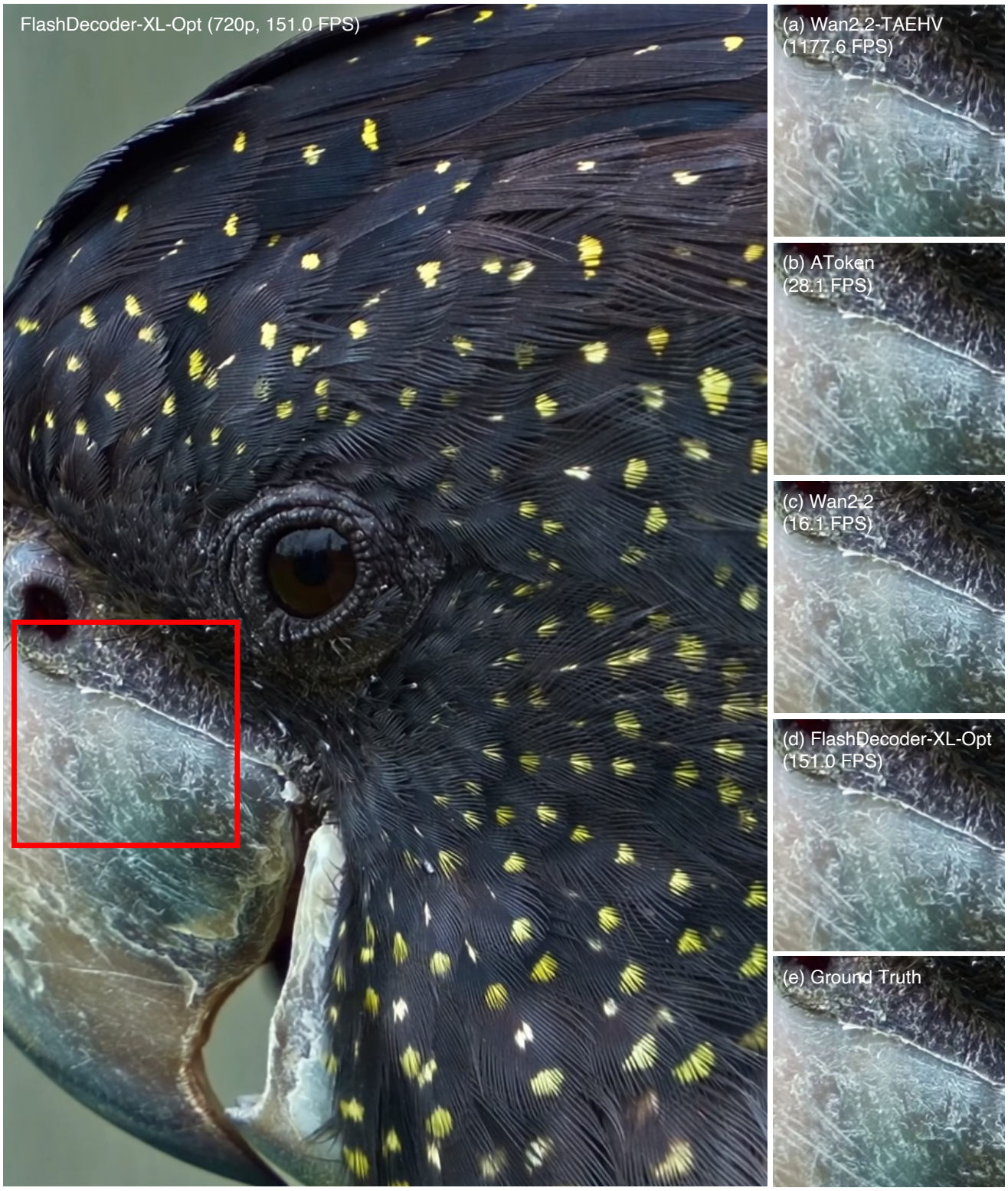

Figure A. **Qualitative comparison of 720p reconstruction results.** We compare reconstructed frames from video decoders with 4× temporal and 16× spatial compression: (a) Wan2.2-TAEHV [1], (b) AToken [13], (c) Wan2.2 [19], (d) our FlashDecoder-XL-Opt, and (e) ground truth. (a) and (b) produce blurry reconstructions, while (c) and (d) yield visually comparable outputs, yet (d) achieves over 9× higher throughput.

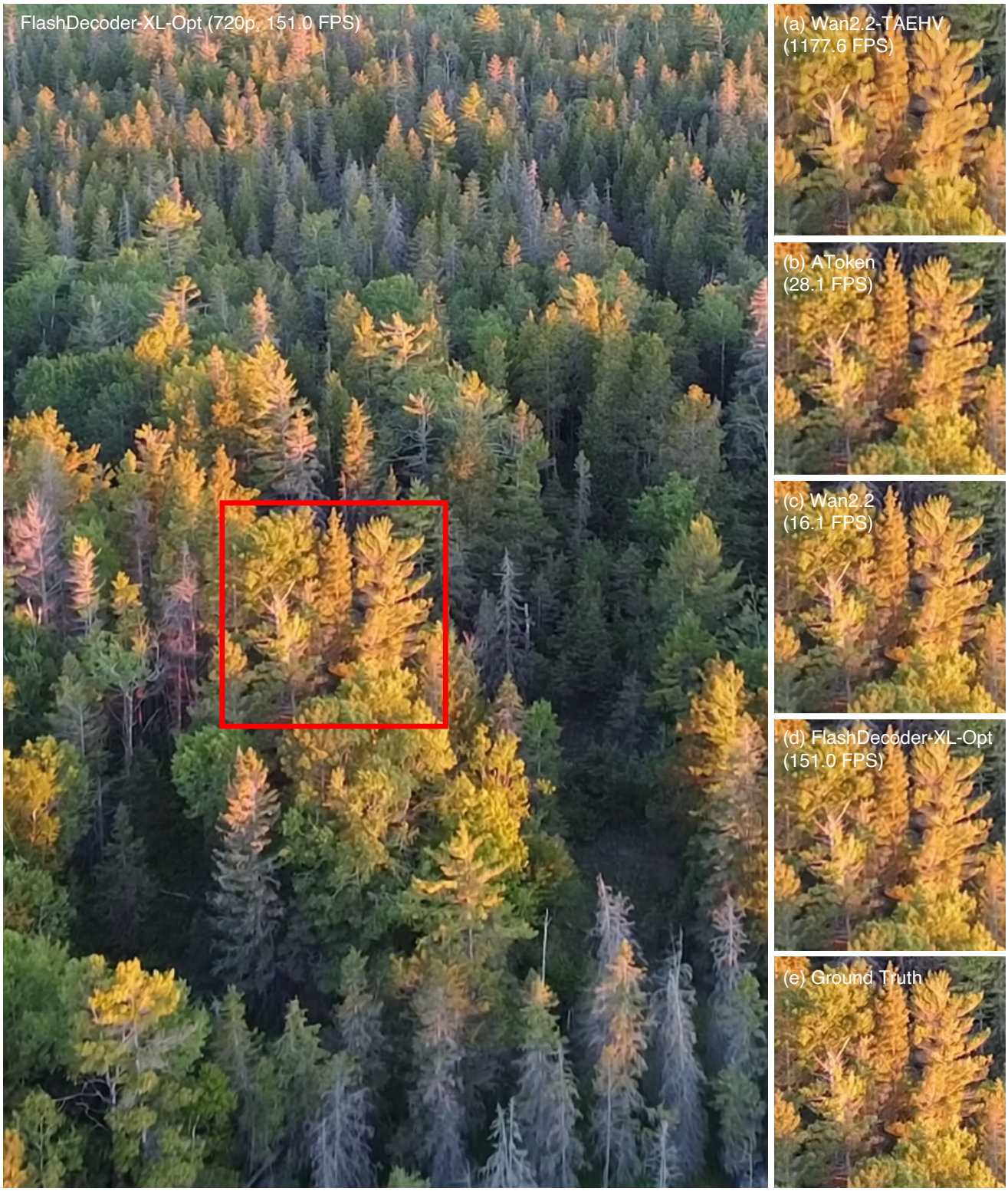

Figure B. **Qualitative comparison of 720p reconstruction results.** We compare reconstructed frames from video decoders with 4× temporal and 16× spatial compression: (a) Wan2.2-TAEHV [1], (b) AToken [13], (c) Wan2.2 [19], (d) our FlashDecoder-XL-Opt, and (e) ground truth. (a) fails to decode fine details such as tree branches and foliage, while (b) produces blurry reconstructions. (c) and (d) yield sharper results, yet (d) achieves over 9× higher throughput.

Figure C. **Qualitative comparison of 720p reconstruction results.** We compare reconstructed frames from video decoders with 4× temporal and 16× spatial compression: (a) Wan2.2-TAEHV [1], (b) AToken [13], (c) Wan2.2 [19], (d) our FlashDecoder-XL-Opt, and (e) ground truth. (a) struggles to decode wall textures near the flowerpot, while (b) produces blurry details in the flowerpot region. (c) and (d) yield visually comparable outputs, with (c) appearing to synthesize marginally finer details, particularly around the flower petals. (d) achieves over 9× higher throughput.

References

[1] Ollin Boer Bohan. Taehv: Tiny autoencoder for hunyuan video. https://github.com/madebyollin/taehv, 2025. 1, 4, 5, 6

[2] Joao Carreira, Eric Noland, Andras Banki-Horvath, Chloe Hillier, and Andrew Zisserman. A short note about kinetics-600. *arXiv preprint arXiv:1808.01340*, 2018. 1

[3] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[4] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. https://github.com/CompVis/taming-transformers, 2021. 1

[5] Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, et al. Datacomp: In search of the next generation of multimodal datasets. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2023. 1

[6] Songwei Ge, Aniruddha Mahapatra, Gaurav Parmar, Jun-Yan Zhu, and Jia-Bin Huang. On the content bias in fréchet video distance. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 3

[7] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2014. 1

[8] Jonathan Heek, Emiel Hoogeboom, Thomas Mensink, and Tim Salimans. Unified latents (ul): How to train your latents. *arXiv preprint arXiv:2602.17270*, 2026. 3

[9] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 2

[10] R. Keys. Cubic convolution interpolation for digital image processing. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 1981. 1

[11] Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024. 3

[12] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations (ICLR)*, 2019. 2

[13] Jiasen Lu, Liangchen Song, Mingze Xu, Byeongjoo Ahn, Yanjun Wang, Chen Chen, Afshin Dehghan, and Yinfei Yang. Atoken: A unified tokenizer for vision. *arXiv preprint arXiv:2509.14476*, 2025. 1, 4, 5, 6

[14] Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. Which Training Methods for GANs do actually Converge? In *International Conference on Machine Learning (ICML)*, 2018. 1, 2

[15] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 3

[16] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025. 3

[17] P Umesh. Image processing in python. *CSI Communications*, 23(2), 2012. 1

[18] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly. Fvd: A new metric for video generation. In *DGS@ICLR*, 2019. 3

[19] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. 4, 5, 6

[20] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. 1, 3

[21] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 2

[22] Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. In *International Conference on Learning Representations (ICLR)*, 2026. 3