Supplementary Material

This supplementary material presents additional implementation details of FlowC2S with further details on the ablation studies, offers extended visual results, and concludes with a discussion of its limitations.

In Appendix A1, we describe the implementation of the method, evaluation setup, the protocol used for the NuScenes dataset [4], and details of the analysis on the effective GPU memory volume.

Appendix A2 reports additional quantitative results; in particular, evaluation using VBench [16] metrics on OpenVid [27] and NuScenes [4], comparison with Bi-Flow [23], and the effect of video chunk size during training.

Appendix A3 provides the algorithms used in the ablation studies, complementing the main algorithm described in the paper.

In Appendix A4, we report additional visual results for both our method and the ablation variants.

Finally, Appendix A5 discusses the limitations of the proposed method and outlines potential directions for future work.

A1. Implementation and Evaluation Details

A1.1. Training Details

The proposed FlowC2S method is fine-tuned from two text-to-video backbones: LTXV [10] and Wan [36]. All ablation studies are run exclusively on LTXV as LTXV's VAE [18] provides a higher overall compression ratio than Wan, yielding faster training and lower compute/memory cost under the same hardware budget (i.e., for the Wan backbone, we double the number of devices relative to LTXV to ensure a matched batch size.).

Unless stated otherwise, we use the AdamW optimizer [25] with a learning rate of 0.0002. The complete set of hyperparameters used for fine-tuning from both LTXV and Wan is listed in Table S1.

A1.2. Evaluation Setup

We evaluate the proposed FlowC2S approach for generating video continuations against world [9, 11] and autoregressive [10, 41] text-to-video methods in a video continuation setting with a fixed number of frames: each model receives 17 conditioning (or input) frames and generates the next 17 frames. The 17/17 choice is driven by the maximum sequence length that fits in memory for Vista [9] (i.e., 34 total frames per sample) on a single NVIDIA H100 GPU with 80 GB of memory. To ensure a fair comparison, we adopt the same 17-frame input and 17-frame output for all methods, including ours. Table S2 reports all evaluation hyperparameters across methods (e.g., input/generation resolution, classifier-free guidance scale [13], and other runtime settings).

A1.2.1. NuScenes Protocol

For the NuScenes dataset [4], we use the validation split from Vista (150 scenes; 750 videos per camera position). We aggregate three camera views – FRONT, BACK, and FRONT-LEFT – and randomly sample 2,000 videos for evaluation to match the sample count of our OpenVid [27] validation set. The exact validation indices for OpenVid and NuScenes are provided with this supplementary material.

A1.3. Details on Effective Volume for GPU Memory Analysis

We compare GPU memory usage across world and autoregressive text-to-video models that rely on different backbones and therefore different VAE compression ratios. To ensure a fair comparison across all methods, we match (or closely approximate) the effective volume per run and per method. Vista and GEM [11] use the SVD [3] backbone; LTXVCondition uses LTXV [10]; CausVid [41] uses Wan [36]; our method is fine-tuned and evaluated with LTXV and Wan backbones.

We characterize the VAE latent space by $c \times f \times h \times w$ (the number of channels, the number of frames, height, and width): SVD $4 \times 1 \times 8 \times 8$, LTXV $128 \times 8 \times 32 \times 32$, Wan $16 \times 4 \times 8 \times 8$. Table S3 lists representative configurations used in the main-text GPU memory study across five effective volumes, including effective volumes for SVD, LTXV, and Wan, and the latent number of conditioning (or input) frames for each run. For consistency, SVD-based methods are run at 576×1024 resolution, and LTXV/Wan-based methods at 480×832.

A2. Extended Experimental Results

A2.1. Additional Quantitative Comparisons

A2.1.1. Evaluation Results Using VBench Metrics

This section reports additional quantitative results on OpenVid [27] and NuScenes [4] using the following metrics from VBench [16]: subject consistency, background consistency, and motion smoothness (Table S4). Table S4 also reports efficiency metrics, measuring total NFE and GPU memory scaling; full details are provided in the main paper and in Appendix A1.3.

On OpenVid, FlowC2S (Wan) surpasses CausVid in subject consistency, background consistency, and motion

Configuration	LTXV-based	Wan-based
Batch Size / GPU	64	32
Accumulate Step	8	8
Optimizer	AdamW	AdamW
β_1	0.9	0.9
β_2	0.99	0.99
Learning Rate	0.0002	0.0002
Learning Rate Schedule	Linear	Cosine
Training Steps	1450	1450
Resolution	256×384	240×416
Number of Frames	17, 41	17, 41
Shifting	True	True
Weighting Scheme	Logit Normal	Uniform
Num Layers	28	30
p	0.7	0.7
Pre-trained Model	LTX-Video-2b-v0.9.5	Wan2.1-T2V-1.3B
Sampler	FlowMatchEulerDiscreteScheduler [8]	UniPCMultistepScheduler [44]
Sample Steps	40	50
Classifier-Free Guidance Scale	3.5	5
Device	NVIDIA H100 80 GB ×28	NVIDIA H100 80 GB ×56
Training Strategy	AMP / DDP / BFloat16	AMP / DDP / BFloat16

Table S1. Fine-tuning hyper-parameters used in our experiments.

Method	H	W	In	Out (img/lat)	Out-L	Blk (frames)	Total NFE	CFG	Rnd
Vista	576	1024	17	17	17	N.A.	50	2.5	1
GEM	576	576	17	17	17	N.A.	117	1.5	1
CausVid	480	832	17	17	5	5	5	1	N.A.
LTXVCondition	256	256	17	17	3	N.A.	40	3.5	N.A.
Ours (LTXV)	256	384	17	17	3	N.A.	5/6/10	3.5	N.A.
Ours (Wan)	240	416	17	17	5	N.A.	5/6/10	5	N.A.

Table S2. Evaluation hyper-parameters across all methods being compared. H: Height, W: Width, In: the number of conditioning or input frames, Out: the number of output frames, Out-L: the number of output frames in the latent space, Blk: frames per block, NFE: number of function evaluations, CFG: classifier-free guidance scale, Rnd: number of sampling rounds, N.A.: not applicable.

smoothness, while requiring only half the input dimensionality. On NuScenes, the same trend holds for background consistency and motion smoothness, with a negligible drop in subject consistency.

A2.1.2. Comparison with Bi-Flow

Bi-flow [23] augments flow matching with noise perturbations at training, similar to bridge models [2, 28, 32, 47] and learns a joint objective to predict both the velocity and noise. Bi-flow generates a video autoregressively from a single-frame, one frame at a time. Yet modeling an entire video segment by a single frame discards the inter-frame structure that models rely on to produce coherent motion, leading to temporal artifacts that compound over extended sequences.

Table S5 reports a quantitative comparison between Bi-Flow and FlowC2S on the OpenVid and NuScenes validation sets, with both methods trained from the LTXV backbone. Since Bi-Flow operates in a frame-by-frame autoregressive manner, generating a 17-frame continuation requires 85 total NFEs (5 NFEs per frame), whereas FlowC2S produces the entire chunk in a single pass with only 5 NFEs. Despite this substantial reduction in NFEs, FlowC2S out-

#	V_{SVD}	$V_{\text{LTXV/Wan}}$	F_{SVD}	F_{LTXV}	F_{Wan}
1	294,912	299,520	5	41	9
2	368,640	399,360	8	57	13
3	479,232	499,200	10	73	17
4	589,824	599,040	13	89	21
5	884,736	898,560	16	137	33

Table S3. Representative data used in the GPU–memory analysis across five effective volumes. V_{SVD}: the effective volume for methods based on the SVD backbone, $V_{\text{LTXV/Wan}}$: the effective volume for methods that use LTXV or Wan as a backbone, F_{SVD}: the latent number of conditioning frames used in methods based on the SVD backbone, F_{LTXV}: the latent number of conditioning (or input) frames used in methods based on the LTXV backbone and F_{Wan}: the latent number of conditioning (or input) frames used in methods that use the Wan backbone.

performs single-frame Bi-flow across all metrics.

A2.1.3. Effect of Chunk Size

We validate the importance of multi-frame video chunks in FlowC2S by ablating the chunk size L used during training, including the single-frame case ($L = 1$). All models use the LTXV backbone, are trained on OpenVid for 1406 steps, and evaluated with 17 input and 17 generated frames at inference for a controlled comparison.

Results are summarized in Tab. S6. Increasing L consistently improves across all metrics, demonstrating that multi-frame temporal context is essential for high-quality and efficient video continuation – a single frame is insufficient to capture the inter-frame video structure.

A2.2. Per-Category Analysis of TI

To analyze the influence of TI on generated video quality, this section reports FID and FVD across NFEs on the categorized OpenVid [27] validation set. First, we detail the categorization procedure; then we present the full per-category results. We categorize our OpenVid validation set along two axes: motion intensity and camera motion type, yielding a 12 categories.

Motion intensity is determined by the motion score provided in the OpenVid metadata, which quantifies the magnitude of scene and object motion. We partition videos into the following three equally-sized groups: slow, medium, and fast (the 33rd and 67th percentiles of the motion score distribution serve as boundaries).

Camera motion type is also taken from the metadata of OpenVid. We define four groups based on keyword matching: (i) static—no camera movement; (ii) pan/tilt—horizontal or vertical camera rotation; (iii) zoom—focal length changes; and (iv) complex—two or more simultaneous camera motion types (e.g., pan with zoom).

Figures S1 and S2 isolate the contribution of TI by comparing w/ inherent OC, w/o TI (blue) against w/ inherent OC, w/ TI (red) across all twelve categories. We make the following observations. First, the benefit of TI is strongly conditioned on camera motion type, and FID and FVD respond to TI differently.

Second, FID values remain consistent across most categories, with the exception of Slow + Zoom, which yields a higher FID than the remaining categories; in this case, w/ inherent OC, w/ TI achieves lower FID than w/ inherent OC, w/o TI. FVD values remain consistent across static camera categories, but are higher under zoom. The following paragraphs discuss each category in detail.

Static categories. Across all three motion speeds, w/ TI and w/o TI are nearly indistinguishable in both FID and FVD. This indicates that TI provides no measurable benefit when inter-chunk displacement is negligible. Under static camera conditions, the temporal coupling alone provides a sufficiently well-conditioned source distribution, and the vector field near the timestep zero requires no additional calibration.

Pan/Tilt categories.TI helps most in this category, on both metrics. In FID, red lies below blue across all speeds, with the gap most pronounced in Slow + Pan/Tilt. In FVD, the same ordering holds: red is below blue and remains stable across NFE, while blue shows a mild upward drift at higher NFE. Pan/Tilt motion produces globally translated but otherwise predictable target latents — the regime where we hypothesize that structured camera motion produces more predictable target latents, making the inverted initialization more informative than in other categories.

Zoom categories. In FID, red is generally below blue — the benefit of TI is present but moderate, and in Slow + Zoom the two curves actually cross, with blue outperforming red at intermediate NFE before red recovers at higher NFE. In FVD, w/o TI degrades with increasing NFE in Medium + Zoom and Fast + Zoom, while w/ TI remains comparatively stable or improves slightly.

Complex categories. In FID, w/ TI is at or below blue across all speeds, with the gap widening for medium and fast speeds. In FVD, w/ TI and w/o TI trade positions across NFE in several panels, most visibly in Fast + Complex, where neither curve dominates cleanly.

The complex category represents a mix of multiple camera motion types. We hypothesize that the predictability assumption underlying TI, that structured motion con-

Method	Visual Quality						Efficiency		
	OpenVid			NuScenes			Total NFE ↓	k (MB / 10^6) ↓	b (MB)
	Subj. Cons.↑	Bg. Cons.↑	Mot. Smooth.↑	Subj. Cons.↑	Bg. Cons.↑	Mot. Smooth.↑			
CausVid [41]	97.30	96.31	99.09	**94.25**	94.16	97.85	5	93.35	4003.58
Ours	**98.37**	**98.43**	**99.29**	92.89	**95.77**	**98.25**	5	**48.64**	3998.74

Table S4. Quantitative comparison on OpenVid and NuScenes (val) using VBench Metrics. Subj.: Subject, Bg.: Background, Cons.: Consistency, Mot.: Motion, Smooth.: Smoothness.

Method	OpenVid			NuScenes		
	FID ↓	FVD ↓	LPIPS_s ↓	FID ↓	FVD ↓	LPIPS_s ↓
Bi-flow [23]	3.34	674.12	0.34	3.99	873.96	0.53
Ours	**0.48**	**129.20**	**0.23**	**1.13**	**185.48**	**0.41**

Table S5. Quantitative comparison with Bi-flow on OpenVid and NuScenes (val) using the LTXV backbone. FlowC2S is reported at 5 NFE. Lower is better (↓); best scores in bold.

L	FID ↓	FVD ↓	LPIPS_s ↓
1	6.81	1390.84	0.39
17	0.52	146.94	0.31
41	**0.50**	**126.54**	**0.24**

Table S6. Impact of chunk size L at training on OpenVid. All models use the LTXV backbone with 5 NFEs. Lower is better (↓); best scores in bold.

strains the source distribution, is weakest here, since complex motion produces less predictable inter-segment transitions. The partial FVD benefit nevertheless suggests TI still improves quality on average, even if the temporal coherence benefit is less stable than under structured camera types.

A3. Ablation Algorithms

In addition to the main training algorithm described in the paper, in this section, we include two auxiliary variants used in our ablation studies for the assessment of the primary design decisions of FlowC2S. Specifically, we provide pseudocode for: fine-tuning without Inherent Optimal Couplings (OC) and without Target Inversion (TI) (Algorithm S1); and fine-tuning with Inherent OC but without Target Inversion (Algorithm S2).

In Algorithm S1, current and succeeding frames are sampled independently from their respective distributions, with no target inversion applied. By contrast, Algorithm S2 incorporates Inherent OC, ensuring coupling between current and succeeding frame chunks as described in the main text, while still omitting target inversion.

A4. Additional Visual Results

This section presents additional qualitative results of generated continuations from FlowC2S (Fig. S3), visual comparisons from ablations on its principal design choices (Fig. S4), as well as studies varying the number of neural function evaluations (NFEs) (Fig. S5) and the number of frames in the generated videos (Fig. S6). The current video chunks are taken from the OpenVid validation set, and the continuations are generated with our model, fine-tuned from LTXV. We set the number of input and generated frames to 41 and resolution to 256×384 in the examples provided in Fig. S3, Fig. S4, and Fig. S5.

Fig. S3 shows the input frames and the generated continuations by our method. FlowC2S produces video continuations that exhibit detailed and strong temporal coherence with the observed context; all achieved without explicit conditioning mechanisms, but simply by taking the current frames directly as an input to the model.

For instance, the motion of the boat traversing left to right with accompanying water dynamics is reproduced with coherent detail. In another case, the car advancing from the background toward the foreground continues seamlessly in the generated continuation, maintaining consistency with both camera dynamics and temporal flow. Thus, FlowC2S preserves logical structure and temporal realism in generated videos.

Fig. S4 presents the current frames as input and the results produced by the model trained by the following four configuration variants: training from scratch with Inherent OC and TI, fine-tuning from LTXV without OC and TI, fine-tuning with OC but without TI, and fine-tuning with both OC and TI.

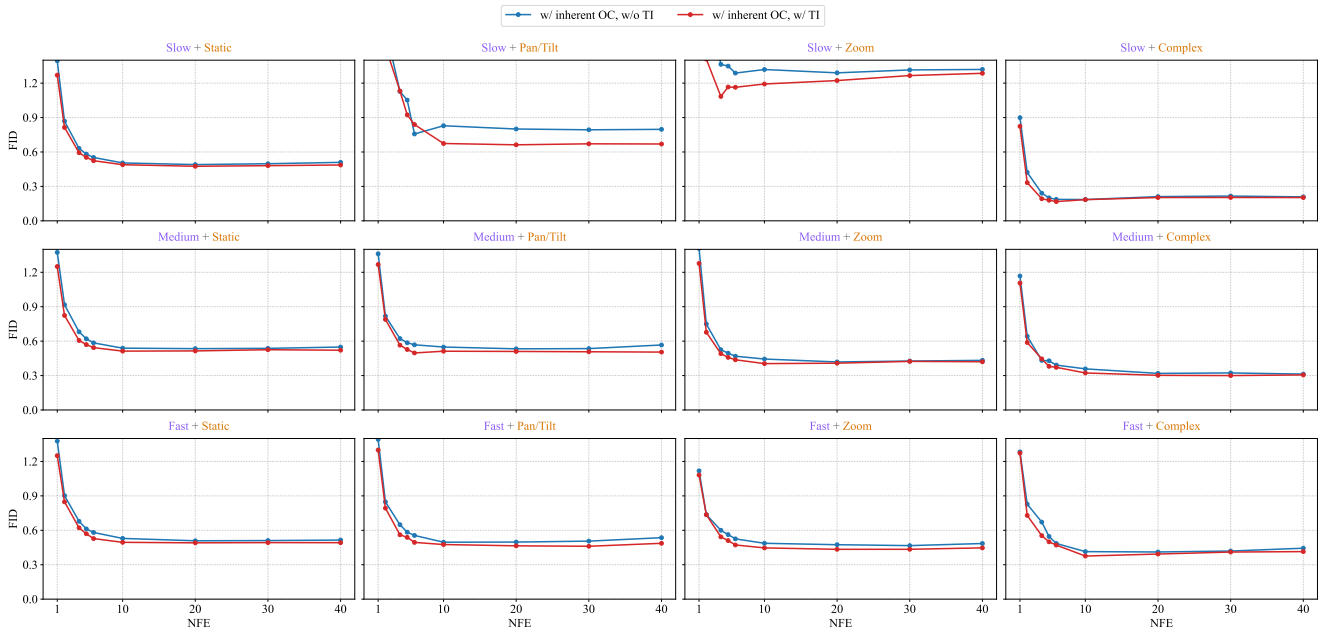

Figure S1. Per-category FID vs. NFE comparing w/ inherent OC, w/o TI (blue) and w/ inherent, OC w/ TI (red). The benefit of TI is strongest under Pan/Tilt camera motion and Fast+Zoom and negligible under static camera.

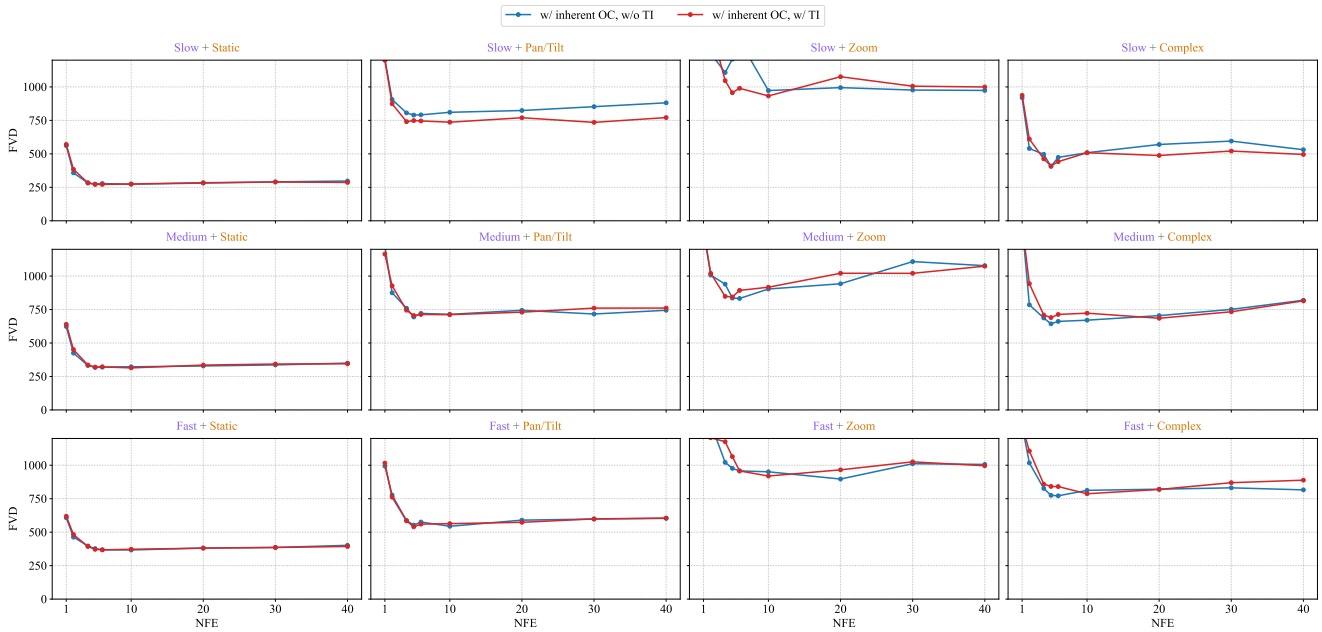

Figure S2. Per-category FVD vs. NFE comparing w/ inherent OC, w/o TI (blue) and w/ inherent, OC w/ TI (red). FVD is substantially higher under zoom than other camera types. Under a static camera, the two variants are indistinguishable.

Training from scratch with OC and TI leads to blurring and visual artifacts and close similarity with the provided input frames, reflecting poor convergence and the significance of using a pre-trained model as an initialization. Similarly, fine-tuning without OC and TI produces unstable results, with noticeable artifacts and color discrepancies (e.g., in the boat and car examples). By contrast, incorporating both OC and TI during fine-tuning yields markedly improved visual fidelity. For instance, in the sea-and-mountain example, the inherent OC-only setting generates blurred

Algorithm S1 Flowing From Current To Succeeding Frames (w/o Inherent OC, w/o Target Inversion)

Require: pretrained $u_t^{\theta^*}$
$u_t^{\theta} \leftarrow u_t^{\theta^*}$
for $x_0 \sim p_{\text{c\_fc}}, x_1 \sim p_{\text{s\_fc}}$ **do**
  $\mu_1, \sigma_1 = \text{VAE}(x_1), \quad x_1 \leftarrow \mu_1$
  $\mu_0, \sigma_0 = \text{VAE}(x_0), \quad x_0 \leftarrow \mu_0$
  $t \sim \mathcal{U}[0, 1]$
  $x \leftarrow (1-t)x_0 + tx_1$
  $\mathcal{L}_{\text{CFM}}(\theta) = \left\| u_t^{\theta}(x) - (x_1 - x_0) \right\|^2$
  Update θ using GD on $\mathcal{L}_{\text{CFM}}(\theta)$
end for

Algorithm S2 Flowing From Current To Succeeding Frames (w/ Inherent OC, w/o Target Inversion)

Require: pretrained $u_t^{\theta^*}$, $\Pi(p_{\text{c\_fc}}, p_{\text{s\_fc}})$
$u_t^{\theta} \leftarrow u_t^{\theta^*}$
for $x_0, x_1 \sim \Pi$ **with** (x_0, x_1) **inherent optimal couplings do**
  $\mu_1, \sigma_1 = \text{VAE}(x_1), \quad x_1 \leftarrow \mu_1$
  $\mu_0, \sigma_0 = \text{VAE}(x_0), \quad x_0 \leftarrow \mu_0$
  $t \sim \mathcal{U}[0, 1]$
  $x \leftarrow (1-t)x_0 + tx_1$
  $\mathcal{L}_{\text{CFM}}(\theta) = \left\| u_t^{\theta}(x) - (x_1 - x_0) \right\|^2$
  Update θ using GD on $\mathcal{L}_{\text{CFM}}(\theta)$
end for

sand textures, while the joint inherent OC and TI setting restores sharper, more realistic details, as highlighted by red boxes in Fig. S4.

Fig. S5 illustrates the input frames and the videos generated by the proposed FlowC2S method with 1, 5, 6, 10, and 40 NFEs. As shown, a single NFE is inadequate for generating video continuations, yielding results with severe motion blur (e.g., in the white car sequence). In contrast, using 5–10 NFEs produces video continuations of competitive fidelity to that obtained with 40 NFEs, demonstrating that our method achieves good visual quality with reduced NFEs.

Fig. S6 depicts the input frames and generated video results of generating long video continuations. In this example, the number of current and generated frames by the model is set to 113. Although FlowC2S is trained only on 17- and 41-frame video chunks, it generalizes to substantially longer video continuations. For example, in the tractor sequence, its generated motion from backward to forward constitutes a plausible temporal continuation with respect to the provided input frames.

A5. Limitations and Future Work

FlowC2S delivers a 2× reduction in input dimensionality and achieves state-of-the-art video continuation in quantitative metrics with substantially fewer NFEs, yet important limitations remain. Below, we elaborate on the constraints outlined in the main text and describe corresponding avenues for future work.

Complex Scenes and Motions. While quantitative metrics (FID and FVD) and the visual results presented in the main text indicate that our method generates plausible video continuations for a diverse set of input frames (including natural, cinematic, and human-centric scenes), the model struggles to handle highly complex motions and abrupt scene transitions. This limitation primarily arises from our dataset construction pipeline: to mitigate abrupt scene changes, we employ a histogram-based scene change detector in order to segment videos into temporally coherent chunks, thereby simplifying the training distribution and enabling evaluation under restricted computational and data budgets.

Therefore, future work will explore training the proposed method on less constrained datasets that retain complex scenes with abrupt transitions, with the goal of improving the model's generalization and robustness to challenging motion and scene dynamics.

Heterogeneous Video Chunk Lengths. In the main text, we showed that FlowC2S maintains stable FID and FVD and strong visual quality for long video continuation with visuals in Fig. S6 with 113 current and generated frames; despite training solely on 17- and 41-frame chunks. Beyond this range, however, results become degraded with visible interpolation artifacts with the input frames. Fig. S7 illustrates representative failure cases at longer video continuation (129 input and generated frames).

Consistent with prior evidence in text-to-image [6, 8, 30, 38] and text-to-video [5, 10, 19, 29, 31, 36, 40, 46, 48] literature, exposure to diverse spatial and temporal scales during training improves inference-time robustness of diffusion and flow-based models. Thus, a natural next step is therefore to train the proposed FlowC2S approach with heterogeneous video chunk lengths, broadening temporal coverage of the model and strengthening generalization for the generation of very long video continuations.

Controllability. While the proposed method introduces a new perspective on video continuations, flowing directly from current to succeeding frames, yielding both a twofold reduction in input dimensionality and fewer neural function evaluations compared to existing methods, the current framework does not leverage additional conditioning signals such as text prompts, depth maps, motion or camera trajectories.

After the arrival of diffusion [7, 14, 33–35] and flow [1, 22, 24] models, controllability has become a key in im-

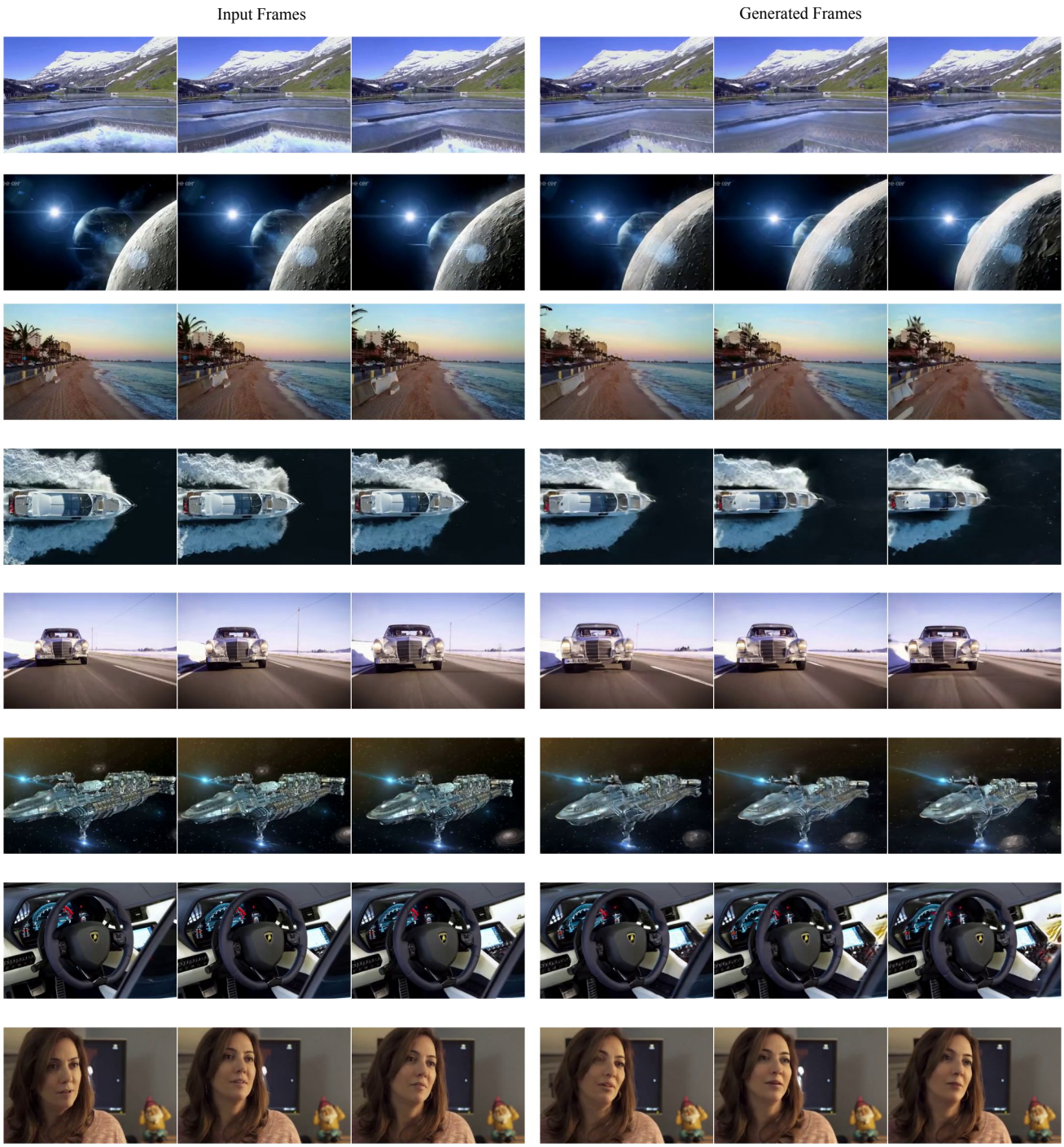

Figure S3. Additional visual results on OpenVid (val). FlowC2S, fine-tuned from LTXV, generates video continuations that are both temporally coherent and visually plausible, maintaining consistency with the content of the input frames. Frames shown at stride 20.

age and video generation [12, 15, 17, 20, 21, 26, 37, 39, 42, 43, 45, 49]. Extending FlowC2S to incorporate auxiliary signals is therefore a promising direction for future work, enabling richer, more adaptable, and user-guided generation of video continuations; in particular, augmenting the current architecture with conditioning signals including scene geometry (e.g., depth or normal maps), object-level motion cues, and camera trajectories could provide fine-

Figure S4. Additional visual results on ablation across four training setups (frames shown with stride 13). Training from scratch with OC+TI introduces visual artifacts, manifested as noticeable interpolation effects between the given input frames and the generated outputs. Fine-tuning without OC+TI results in generated videos with temporal inconsistencies with the current frames and color shifts. Incorporating both OC and TI during fine-tuning yields sharper details than OC-only fine-tuning (see red boxes).

grained spatial and temporal control over the synthesized content and better align the generated video continuations with user intent and downstream application constraints.

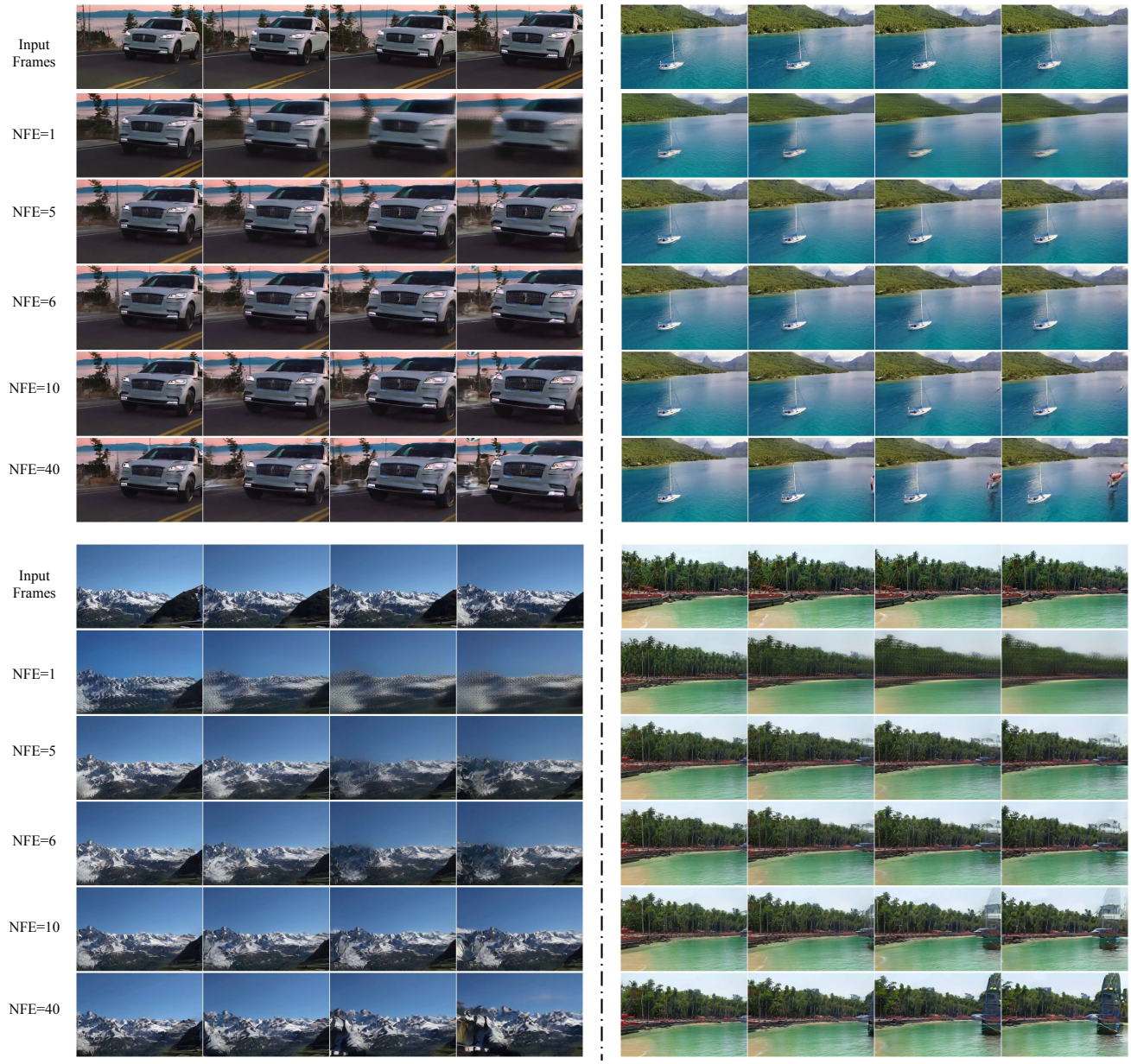

Figure S5. Ablation on neural function evaluations (NFEs). Frames are shown with a stride of 13. 5–10 NFEs yield quality comparable to 40 NFEs, whereas a single NFE produces blurry, degraded outputs.

Figure S6. Long video continuation. The number of input and future frames is 113, and the frames are visualized with a stride of 28. Despite being trained only on 17- and 41-frame sequences, FlowC2S successfully generates coherent long-video continuations.

Figure S7. Failure cases for very long continuation. Shown are 129 input and generated frames (visualized with a stride of 28). Beyond 113-frame video chunks, FlowC2S degrades, exhibiting interpolation artifacts and reduced coherence with the given context.

References

[1] Michael S. Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants, 2023. 6

[2] Michael S. Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions, 2025. 2

[3] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, Varun Jampani, and Robin Rombach. Stable video diffusion: Scaling latent video diffusion models to large datasets, 2023. 1

[4] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving, 2020. 1

[5] Shoufa Chen, Chongjian Ge, Yuqi Zhang, Yida Zhang, Fengda Zhu, Hao Yang, Hongxiang Hao, Hui Wu, Zhichao Lai, Yifei Hu, Ting-Che Lin, Shilong Zhang, Fu Li, Chuan Li, Xing Wang, Yanghua Peng, Peize Sun, Ping Luo, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Xiaobing Liu. Goku: Flow based video generative foundation models, 2025. 6

[6] Xiaoliang Dai, Ji Hou, Chih-Yao Ma, Sam Tsai, Jialiang Wang, Rui Wang, Peizhao Zhang, Simon Vandenhende, Xiaofang Wang, Abhimanyu Dubey, Matthew Yu, Abhishek Kadian, Filip Radenovic, Dhruv Mahajan, Kunpeng Li, Yue Zhao, Vladan Petrovic, Mitesh Kumar Singh, Simran Motwani, Yi Wen, Yiwen Song, Roshan Sumbaly, Vignesh Ramanathan, Zijian He, Peter Vajda, and Devi Parikh. Emu: Enhancing image generation models using photogenic needles in a haystack, 2023. 6

[7] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *Advances in Neural Information Processing Systems*, pages 8780–8794. Curran Associates, Inc., 2021. 6

[8] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis, 2024. 2, 6

[9] Shenyuan Gao, Jiazhi Yang, Li Chen, Kashyap Chitta, Yihang Qiu, Andreas Geiger, Jun Zhang, and Hongyang Li. Vista: A generalizable driving world model with high fidelity and versatile controllability. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 1

[10] Yoav HaCohen, Nisan Chiprut, Benny Brazowski, Daniel Shalem, Dudu Moshe, Eitan Richardson, Eran Levin, Guy Shiran, Nir Zabari, Ori Gordon, Poriya Panet, Sapir Weissbuch, Victor Kulikov, Yaki Bitterman, Zeev Melumian, and Ofir Bibi. Ltx-video: Realtime video latent diffusion, 2024. 1, 6

[11] Mariam Hassan, Sebastian Stapf, Ahmad Rahimi, Pedro M B Rezende, Yasaman Haghighi, David Brüggemann, Isinsu Katircioglu, Lin Zhang, Xiaoran Chen, Suman Saha, Marco Cannici, Elie Aljalbout, Botao Ye, Xi Wang, Aram Davtyan, Mathieu Salzmann, Davide Scaramuzza, Marc Pollefeys, Paolo Favaro, and Alexandre Alahi. Gem: A generalizable ego-vision multimodal world model for fine-grained ego-motion, object dynamics, and scene composition control, 2024. 1

[12] Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. Cameractrl: Enabling camera control for text-to-video generation, 2025. 7

[13] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022. 1

[14] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, pages 6840–6851. Curran Associates, Inc., 2020. 6

[15] Chen Hou and Zhibo Chen. Training-free camera control for video generation, 2025. 7

[16] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 1

[17] Zeyinzi Jiang, Zhen Han, Chaojie Mao, Jingfeng Zhang, Yulin Pan, and Yu Liu. Vace: All-in-one video creation and editing, 2025. 7

[18] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022. 1

[19] Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, Kathrina Wu, Qin Lin, Junkun Yuan, Yanxin Long, Aladdin Wang, Andong Wang, Changlin Li, Duojun Huang, Fang Yang, Hao Tan, Hongmei Wang, Jacob Song, Jiawang Bai, Jianbing Wu, Jinbao Xue, Joey Wang, Kai Wang, Mengyang Liu, Pengyu Li, Shuai Li, Weiyan Wang, Wenqing Yu, Xinchi Deng, Yang Li, Yi Chen, Yutao Cui, Yuanbo Peng, Zhentao Yu, Zhiyu He, Zhiyong Xu, Zixiang Zhou, Zunnan Xu, Yangyu Tao, Qinglin Lu, Songtao Liu, Dax Zhou, Hongfa Wang, Yong Yang, Di Wang, Yuhong Liu, Jie Jiang, and Caesar Zhong. Hunyuanvideo: A systematic framework for large video generative models, 2025. 6

[20] Guojun Lei, Chi Wang, Rong Zhang, Yikai Wang, Hong Li, and Weiwei Xu. Animateanything: Consistent and controllable animation for video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 27946–27956, 2025. 7

[21] Jingyun Liang, Yuchen Fan, Kai Zhang, Radu Timofte, Luc Van Gool, and Rakesh Ranjan. Movideo: Motion-aware video generation with diffusion models, 2024. 7

[22] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling, 2023. 6

[23] Chen Liu and Tobias Ritschel. Generative video bi-flow. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 19363–19372, 2025. 1, 2, 4

[24] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow, 2022. 6

[25] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. 1

[26] Yue Ma, Kunyu Feng, Zhongyuan Hu, Xinyu Wang, Yucheng Wang, Mingzhe Zheng, Xuanhua He, Chenyang Zhu, Hongyu Liu, Yingqing He, Zeyu Wang, Zhifeng Li, Xiu Li, Wei Liu, Dan Xu, Linfeng Zhang, and Qifeng Chen. Controllable video generation: A survey, 2025. 7

[27] Kepan Nan, Rui Xie, Penghao Zhou, Tiehan Fan, Zhenheng Yang, Zhijie Chen, Xiang Li, Jian Yang, and Ying Tai. Openvid-1m: A large-scale high-quality dataset for text-to-video generation. *arXiv preprint arXiv:2407.02371*, 2024. 1, 3

[28] Stefano Peluchetti. Non-denoising forward-time diffusions, 2023. 2

[29] Xiangyu Peng, Zangwei Zheng, Chenhui Shen, Tom Young, Xinying Guo, Binluo Wang, Hang Xu, Hongxin Liu, Mingyan Jiang, Wenjun Li, Yuhui Wang, Anbang Ye, Gang Ren, Qianran Ma, Wanying Liang, Xiang Lian, Xiwen Wu, Yuting Zhong, Zhuangyan Li, Chaoyu Gong, Guojun Lei, Leijun Cheng, Limin Zhang, Minghao Li, Ruijie Zhang, Silan Hu, Shijie Huang, Xiaokang Wang, Yuanheng Zhao, Yuqi Wang, Ziang Wei, and Yang You. Open-sora 2.0: Training a commercial-level video generation model in $200k$, 2025. 6

[30] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis, 2023. 6

[31] Team Seawead, Ceyuan Yang, Zhijie Lin, Yang Zhao, Shanchuan Lin, Zhibei Ma, Haoyuan Guo, Hao Chen, Lu Qi, Sen Wang, Feng Cheng, Feilong Zuo, Xuejiao Zeng, Ziyan Yang, Fangyuan Kong, Meng Wei, Zhiwu Qing, Fei Xiao, Tuyen Hoang, Siyu Zhang, Peihao Zhu, Qi Zhao, Jiangqiao Yan, Liangke Gui, Sheng Bi, Jiashi Li, Yuxi Ren, Rui Wang, Huixia Li, Xuefeng Xiao, Shu Liu, Feng Ling, Heng Zhang, Houmin Wei, Huafeng Kuang, Jerry Duncan, Junda Zhang, Junru Zheng, Li Sun, Manlin Zhang, Renfei Sun, Xiaobin Zhuang, Xiaojie Li, Xin Xia, Xuyan Chi, Yanghua Peng, Yuping Wang, Yuxuan Wang, Zhongkai Zhao, Zhuo Chen, Zuquan Song, Zhenheng Yang, Jiashi Feng, Jianchao Yang, and Lu Jiang. Seaweed-7b: Cost-effective training of video generation foundation model, 2025. 6

[32] Yuyang Shi, Valentin De Bortoli, Andrew Campbell, and Arnaud Doucet. Diffusion schrödinger bridge matching, 2023. 2

[33] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 2256–2265, Lille, France, 2015. PMLR. 6

[34] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

[35] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations, 2021. 6

[36] 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, 2025. 1, 6

[37] Shitao Xiao, Yueze Wang, Junjie Zhou, Huaying Yuan, Xingrun Xing, Ruiran Yan, Chaofan Li, Shuting Wang, Tiejun Huang, and Zheng Liu. Omnigen: Unified image generation, 2024. 7

[38] Enze Xie, Junsong Chen, Junyu Chen, Han Cai, Haotian Tang, Yujun Lin, Zhekai Zhang, Muyang Li, Ligeng Zhu, Yao Lu, and Song Han. Sana: Efficient high-resolution image synthesis with linear diffusion transformers, 2024. 6

[39] Dejia Xu, Weili Nie, Chao Liu, Sifei Liu, Jan Kautz, Zhangyang Wang, and Arash Vahdat. Camco: Camera-controllable 3d-consistent image-to-video generation, 2024. 7

[40] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, Da Yin, Yuxuan Zhang, Weihan Wang, Yean Cheng, Bin Xu, Xiaotao Gu, Yuxiao Dong, and Jie Tang. Cogvideox: Text-to-video diffusion models with an expert transformer, 2025. 6

[41] Tianwei Yin, Qiang Zhang, Richard Zhang, William T Freeman, Fredo Durand, Eli Shechtman, and Xun Huang. From slow bidirectional to fast autoregressive video diffusion models. 2025. 1, 4

[42] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023. 7

[43] Yabo Zhang, Yuxiang Wei, Dongsheng Jiang, Xiaopeng Zhang, Wangmeng Zuo, and Qi Tian. Controlvideo: Training-free controllable text-to-video generation, 2023. 7

[44] Wenliang Zhao, Lujia Bai, Yongming Rao, Jie Zhou, and Jiwen Lu. Unipc: A unified predictor-corrector framework for fast sampling of diffusion models, 2023. 2

[45] Sixiao Zheng, Zimian Peng, Yanpeng Zhou, Yi Zhu, Hang Xu, Xiangru Huang, and Yanwei Fu. Vidcraft3: Camera, object, and lighting control for image-to-video generation, 2025. 7

[46] Zangwei Zheng, Xiangyu Peng, Tianji Yang, Chenhui Shen, Shenggui Li, Hongxin Liu, Yukun Zhou, Tianyi Li, and Yang You. Open-sora: Democratizing efficient video production for all, 2024. 6

[47] Linqi Zhou, Aaron Lou, Samar Khanna, and Stefano Ermon. Denoising diffusion bridge models, 2023. 2

[48] Yuan Zhou, Qiuyue Wang, Yuxuan Cai, and Huan Yang. Allegro: Open the black box of commercial-level video generation model, 2024. 6

[49] Yujie Zhou, Jiazi Bu, Pengyang Ling, Pan Zhang, Tong Wu, Qidong Huang, Jinsong Li, Xiaoyi Dong, Yuhang Zang, Yuhang Cao, Anyi Rao, Jiaqi Wang, and Li Niu. Light-a-video: Training-free video relighting via progressive light fusion, 2025. 7