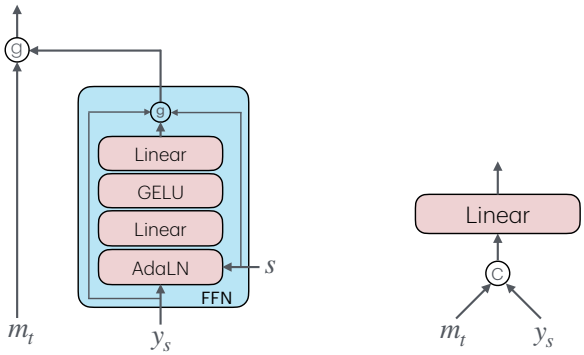


(a) Fusion type: "gated" (b) Fusion type: "concat"

Figure 8. Visualization of our mechanisms to fuse the main backbone's features $\boldsymbol{m}(\boldsymbol{x}, t)$ with the flow head input $\boldsymbol{y}_s$: (a) "gated" fusion, where the flow head input $\boldsymbol{y}_s$ is passed to an FFN block (conditioned on the flow head timestep s) and then added to the main feature $\boldsymbol{m}_t$ via a gated operator; and (b) "concat" fusion, where the the flow head input $\boldsymbol{y}_s$ and the main feature $\boldsymbol{m}_t$ are concatenated in the channel dimension and then passed to the linear projection layer.

Algorithm 1 TMD inference

$\boldsymbol{x} \sim \mathcal{N}(0, I)$
for $i = M$ **to** 1 **do**
  $\boldsymbol{m} = \boldsymbol{m}_\theta(\boldsymbol{x}, t_i)$ ▷ Main backbone
  $\boldsymbol{x} = \boldsymbol{x} - (t_i - t_{i-1})\textsc{InnerFlow}(\boldsymbol{m})$ ▷ See Eq. (4)
return $\boldsymbol{x}$ ▷ Generated data

def $\textsc{InnerFlow}(\boldsymbol{m})$
  $\boldsymbol{y} \sim \mathcal{N}(0, I)$
  for $j = N$ **to** 1 **do**
    $\boldsymbol{y} = \boldsymbol{f}_\theta(\boldsymbol{y}, s_j, s_{j-1}; \boldsymbol{m})$ ▷ Flow head
  return $\boldsymbol{y}$

A. Implementation details

A.1. Flow head conditioning

Time conditioning. For the flow head, we re-use the default time embedding to embed s and instantiate an additional zero-initialized time embedding that embeds the difference $s - r$ between two time steps of the flow map (see Section 2). The two embeddings are summed and used as the conditioning input to the adaptive normalization layers in the DiT blocks of the flow head.

Inner flow fusion. To fuse the main backbone's features $\boldsymbol{m}(\boldsymbol{x}, t)$ with the flow head input $\boldsymbol{y}_s$, we first process $\boldsymbol{y}_s$ with the patch embedding layer of the main backbone. Then we process the patches using a randomly initialized AdaLN-style block (analogous to the ones used in the teacher model, *i.e.*, consisting of adaptive layer normalization, an MLP, and a gated residual connection) conditioned on the embedding of s and re-using the global scale and shift values. Finally, we fuse this processed flow head input with the main backbone's features using a gated interpolation, controlled by a global gate derived from a sigmoid-activated learnable parameter. This makes the flow head's contribution both time-aware and dynamically gated at the feature level before the final global fusion. We visualize the fusion mechanism in Figure 8 and provide an ablation w.r.t. other conditioning types in Section B.3.

A.2. TM-MF

Pretraining recipe. To better align TM-MF pretraining with the multi-step inference strategy used in our distillation setup, we modify the sampling scheme of (s, r) compared to the continuous formulation in [17]. First, analogous to t_{student} in the outer flow of DMD2-*v*, we also use a shifting function with $\gamma = 10$ for s_{student}, i.e., the values defining the time grid[2] $0 = s_0 < s_1 < \cdots < s_N = 1$. Secondly, analogous to sampling the timestep t_{dmd} in DMD2-*v*, we also sample s_{mf} during TM-MF training uniformly and shift it (using $\gamma = 3$). Then, we pick $r = s_k$ where $k := \max\{j : s_j \leq s_{\text{mf}}\}$. Empirically, this discrete sampling scheme is not only tailored to our Stage 2 distillation but also substantially stabilizes TM-MF pretraining.

Our remaining design choices largely follow Geng et al. [17]. We set $r = s$ for 75% of training batches to stabilize optimization under the flow-matching loss, apply condition dropout (using the negative prompt in Table 8) during training, and construct the velocity field using classifier-free guidance. We observed that mixing conditional and unconditional network-predicted velocities (Eq. (18) in [17]) offers no clear benefit in our setting, so we adopt the standard classifier-free guidance formulation throughout. Moreover, we also use an adaptive loss normalization, such that our final loss derived from (9) is given by

$$\mathbb{E}_{s,r,\boldsymbol{y}_s}\left[\frac{\|\boldsymbol{u}_\theta(\boldsymbol{y}_s, s, r) - \hat{\boldsymbol{u}}\|^2}{\text{sg}\left(\|\boldsymbol{u}_\theta(\boldsymbol{y}_s, s, r) - \hat{\boldsymbol{u}}\|^2\right) + c}\right],$$

where we choose $c = d$ for Wan2.1 1.3B and $c = \frac{d}{10^5}$ for Wan2.1 14B (where d is the dimension of $\boldsymbol{y}_s$).

Finite difference approximation. Computing the JVP term in the MeanFlow objective, namely $\frac{\mathrm{d}}{\mathrm{d}s}\boldsymbol{u}_\theta(\boldsymbol{y}_s, s, r)$, using forward-mode automatic differentiation in PyTorch is currently incompatible with system optimizations, such as flash attention and FSDP; however, these optimizations are crucial to avoid out-of-memory issues due to the long video sequence length. To address this and make our method agnostic of different training techniques, we approximate the JVP using a *central difference scheme*. For a step size δ, we

[2]Note that, in practice, we set $t_M = s_N = 0.999$ instead of 1 to align with the pretraining of Wan.

Hyperparameter	Wan2.1 1.3B (T2V)	Wan2.1 14B (T2V)
General Settings		
Resolution	480×832 (480P)	
Frame count	81 Frames (5s)	
Latent dimension $(T \times H \times W)$	$21 \times 60 \times 104$	
Dataset size	500k (479k after filtering)	
Dataset prompts	VidProm extended by QWEN2.5-7B	
Dataset videos	Generated by Wan2.1 14B (T2V)	
Optimizer (weight decay, betas, epsilon)	AdamW $(0.01, (0.9, 0.99), 10^{-8})$	
Global batch-size	64	
Timestep shift γ **for** t_{student} (see Section 3.2)	10	
Multi-step sampling (see Section A.3)	deterministic	
Precision	BF16 (time t in FP64)	
GPU type	NVIDIA A100	NVIDIA H100
Parallelism Strategy	DDP	FSDP
Baselines		
KD		
Number of trajectories	10k	
CFG scale	5	
Skip-layer	10	
Learning rate	7e-5	
Max. iterations	10k	5k
DMD2-*v*		
Fake score architecture	same as teacher	
Discriminator architecture (see Section 4.3)	Conv3D	
Discriminator parameters	68M	172M
CFG scale	5	
Layer for discriminator features (see Section A.3)	$(15, 22, 29)$	$(19, 29, 39)$
Discriminator loss weight λ (see Algorithm 2)	0.03	
Learning rates (student, discriminator, fake score)	$(10^{-5}, 10^{-5}, 10^{-5})$	
Timestep shift γ **for** t_{dmd} (see Section 3.2)	5	
Min./max. for t_{dmd} **and** s_{mf}	$(0.001, 0.999)$	
Student update frequency	every 5-th iteration	
Max. iterations	4k $(M \geq 2)$, 12k $(M = 1)$	1k $(M \geq 2)$, 5k $(M = 1)$
TMD		
Stage 1: TM-MF		
CFG scale	3	
Flow head fuse type (see Section A.1)	gated	
Timestep shift γ **for** s_{student} (see Section A.2)	10	
Timestep shift γ **for** s_{mf} (see Section A.2)	3	
JVP finite-difference δ (see Section A.2)	$5 \cdot 10^{-3}$	
Condition dropout probability (see Section A.2)	0.1	
Loss normalization constant c (see Section A.2)	d	$10^{-5}d$
Learning rate	$3 \cdot 10^{-5}$	$1 \cdot 10^{-5}$
Max. iterations	3k	3k
Stage 2: DMD2-*v* with flow head		
All hyperparameters as in DMD2-*v* above		

Table 7. Default hyperparameters if not specified otherwise in the experiments.

Algorithm 2 TMD student update step (simplified)

Given $\boldsymbol{x} \sim p_{\text{data}},\ \boldsymbol{x}_1 \sim \mathcal{N}(0, I),\ t_i \sim \text{Unif}(\{t_1, \ldots, t_M\})$
$\boldsymbol{x}_{t_i} = (1 - t_i)\boldsymbol{x} + t_i \boldsymbol{x}_1$ ▷ See Eq. (1)
$\boldsymbol{m} = \boldsymbol{m}_\theta(\boldsymbol{x}_{t_i}, t_i)$ ▷ Main backbone
if stage_one **then**
  $\boldsymbol{y} = \boldsymbol{x}_1 - \boldsymbol{x}$ ▷ See Eq. (3)
  $\boldsymbol{y}_1 \sim \mathcal{N}(0, I),\ (s, r) \sim p_{s,r}$
  $\boldsymbol{y}_s = (1 - s)\boldsymbol{y} + s\boldsymbol{y}_1$ ▷ See Eq. (5)
  $\boldsymbol{u} = \boldsymbol{u}_\theta(\boldsymbol{y}_s, s, r; \boldsymbol{m})$ ▷ Avg. velocity
  $\boldsymbol{v} = \boldsymbol{y}_1 - \boldsymbol{y}$ ▷ Conditional velocity
  $\mathcal{L} = \text{MeanFlow}(\boldsymbol{u}, \boldsymbol{v}, s, r)$ ▷ See Eq. (9)
else
  $\hat{\boldsymbol{x}} = \boldsymbol{x}_1 - \textsc{InnerFlow}(\boldsymbol{m})$ ▷ See Eq. (15) & Algorithm 1
  $\mathcal{L} = \text{VSD}(\hat{\boldsymbol{x}}) + \lambda \cdot \text{Discriminator}(\hat{\boldsymbol{x}})$ ▷ See Eq. (11)
$\theta = \text{step}(\theta, \nabla_\theta \mathcal{L})$ ▷ Gradient step

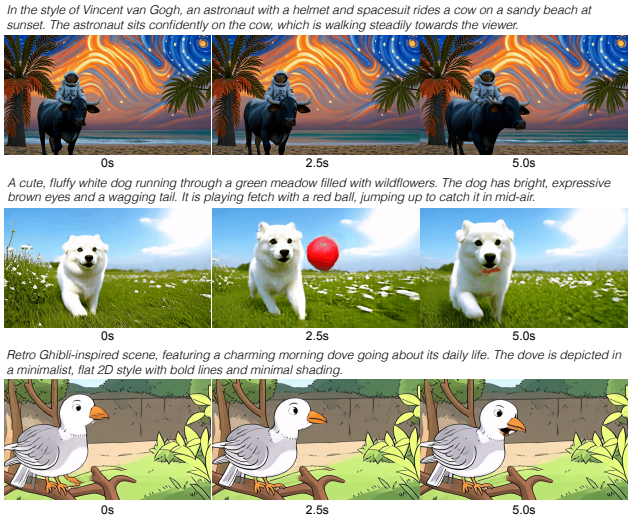


Figure 9. **Mode collapse without time-shifting.** We show videos generated by the one-step student distilled from DMD2 in the setting "t_{dmd} w/o shift" (see Table 5), where the other hyperparameters use the default values. We can see that all generated videos have the main characters consistently appear on the left side of the pixel space, which is a sign of the severe mode collapse happening during distillation.

have

$$\frac{\mathrm{d}}{\mathrm{d}s}\boldsymbol{u}_\theta(\boldsymbol{y}_s, s, r) \approx \frac{\boldsymbol{u}_\theta(\boldsymbol{y}_{s+\delta}, s+\delta, r) - \boldsymbol{u}_\theta(\boldsymbol{y}_{s-\delta}, s-\delta, r)}{2\delta}$$

with $\boldsymbol{y}_{s\pm\delta} = \boldsymbol{y}_s \pm \delta \boldsymbol{v}(\boldsymbol{y}_s, s)$, where we use the conditional velocity for $\boldsymbol{v}$. At the boundaries, we fall back to using one-sided finite-difference approximations. We empirically found that setting $\delta = 0.005$ yields a satisfactory estimation to the JVP term and fix it throughout our experiments.

A.3. DMD2

Initialization. Since we consider video diffusion models that are trained to approximate the instantaneous velocity $\boldsymbol{v}$ in (2), we parametrize the student network as

$$\boldsymbol{g}_\theta^{\text{student}}(\boldsymbol{x}_t, t) = \boldsymbol{x}_t - t\boldsymbol{v}_\theta(\boldsymbol{x}_t, t) \tag{17}$$

for our DMD2-v baseline, which initially approximates the conditional expectation $\mathbb{E}[\boldsymbol{x}|\boldsymbol{x}_t]$.

Multi-step distillation and inference. Compared to the original multi-step DMD2 [58], we did not use backward simulation for the outer loop for multi-step distillation and efficiently sample $\boldsymbol{x}_{t_i}$ using real data.

Moreover, during inference, we use deterministic sampling from the conditional flow, *i.e.*,

$$\boldsymbol{x}_{t_{i+1}} = \left(1 - \frac{t_{i+1}}{t_i}\right)\boldsymbol{x}_{t_i} + \frac{t_{i+1}}{t_i}\boldsymbol{g}^{\text{student}}(\boldsymbol{x}_{t_i}, t_i), \tag{18}$$

which samples from the correct distribution as long as $\boldsymbol{x} \approx \text{Student}(\boldsymbol{x}_{t_i}, t_i)$. Compared to the standard resampling scheme in DMD2, *i.e.*,

$$\boldsymbol{x}_{t_{i+1}} = (1 - t_{i+1})\,\boldsymbol{g}^{\text{student}}(\boldsymbol{x}_{t_i}, t_i) + t_{i+1}\boldsymbol{x}_1, \tag{19}$$

the noise $\boldsymbol{x}_1$ of the conditional flow is inferred from $\boldsymbol{g}^{\text{student}}(\boldsymbol{x}_{t_i}, t_i)$ and $\boldsymbol{x}_{t_i}$ in Eq. (18) and sampled independently in Eq. (19).

For TMD, the multi-step inference is given in Algorithm 1. In particular, the outer transitions are also deterministic, but additional independent noise $\boldsymbol{y}_1 \sim \mathcal{N}(0, I)$ is used for the inner flow.

Teacher, fake score, and discriminator. While it would be possible to distill Wan2.1 1.3B using the 14B model as teacher in DMD2, we use the 1.3B model for fair comparisons. Moreover, we use (unpatchified) teacher features at different layers as inputs to the discriminator, which we found more stable than using fake score features. The discriminator is trained using an average minimax log-likelihood objective over separate heads for each teacher feature, where the teacher is evaluated at noisy inputs $(\hat{\boldsymbol{x}}_t, t)$ (fake data as defined in Section 2) or $(\boldsymbol{x}_t, t)$ (real data). The generator loss is given as the average (non-saturating) negative log-likelihood. We initialize the fake score $\boldsymbol{g}^{\text{fake}}$ using the teacher parameters, parametrize it analogous to (17), and use denoising score matching to train it on noisy fake data $(\hat{\boldsymbol{x}}_t, t)$. Both the discriminator and fake score are trained for several iterations in between student updates. Finally, we can write our VSD objective in (11) as

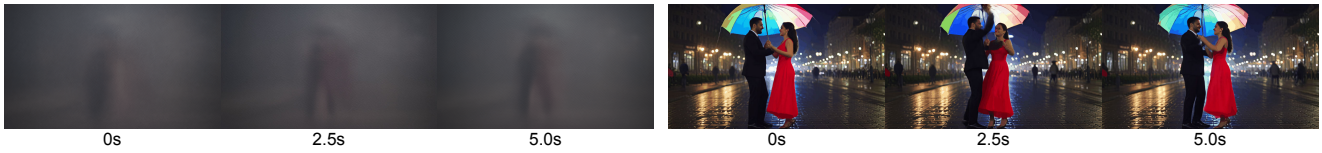
$$\mathbb{E}_{t_i, \boldsymbol{x}_{t_i}, t, \hat{\boldsymbol{x}}_t}\left[\text{sg}\left(\frac{\boldsymbol{g}^{\text{fake}}(\hat{\boldsymbol{x}}_t, t) - \boldsymbol{g}^{\text{teacher}}(\hat{\boldsymbol{x}}_t, t)}{\|\boldsymbol{g}^{\text{fake}}(\hat{\boldsymbol{x}}_t, t) - \boldsymbol{g}^{\text{teacher}}(\hat{\boldsymbol{x}}_t, t)\|_1}\right)^T \hat{\boldsymbol{x}}\right],$$

where $\text{sg}(\cdot)$ denotes the stop-gradient operation, and the teacher is also parametrized analogous to Eq. (17), including CFG with the negative prompt given in Table 8.

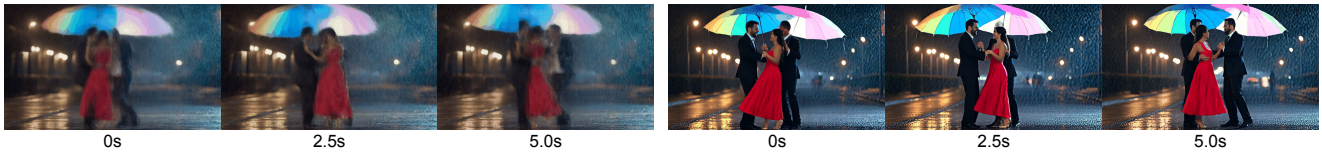
A.4. VBench evaluation

We follow the official evaluation protocol in VBench [22] to test our method. During video sampling, we use standard VBench prompt lists to ensure fair comparisons among different methods. In particular, we rewrite the test prompts (*i.e.*, prompt augmentation) using QWEN/QWEN2.5-7B-

A romantic scene featuring a happy couple dancing salsa in the rain under a large colorful umbrella. They are dressed in elegant evening attire, with the man in a black suit and the woman in a flowing red dress. Both have joyful expressions as they move gracefully to the music, their feet barely touching the wet pavement. Raindrops fall softly around them, creating a misty atmosphere.



(a) Two-step DMD2 w/o KD warm-up (*Left*: Iteration 0, *Right*: Iteration 1000)



(b) Two-step DMD2 w/ KD warm-up (*Left*: Iteration 0, *Right*: Iteration 1000)

Figure 10. **Effect of KD initialization.** We compare the two-step DMD2 results of distilling Wan2.1 1.3B in two settings: (a) with and (b) without KD warm-up, where (left) "Iteration 0" means videos generated in the beginning of DMD2 training and (Right) "Iteration 1000" means videos generated after training DMD2 for 1k iterations. From left to right, we show the first, middle and the last frames in each video. We can see that the KD warm-up initially can generate better videos, but it also introduces coarse-grained artifacts. For instance, it generates an extra man besides a couple specified in the prompt. After training for 1k iterations, two-step DMD2 cannot remove these artifacts, leading to the worse generation quality than two-step DMD2 without KD warm-up.

INSTRUCT [55], following the prior work [21]. Similarly, almost all the baselines, including Wan2.1 base models, also enhance VBench prompts, making them longer and more descriptive without altering their original meaning. We then calculate scores for 16 Text-to-Video (T2V) evaluation dimensions, respectively, and summarize them into quality score, semantic score and overall score.

In figures from the main text, we only show the short prompts due to the space limit. In Table 8, we show the corresponding extended prompts in Figures 1-4 that are actually passed to the model.

A.5. Hyperparameters

We provide an overview of our default hyperparameters in Table 7.

B. Additional experiments

B.1. Quality-efficiency tradeoff

Extending the experiments in Section 4.3, Figure 11 shows TMD's performance-efficiency tradeoff when varying both the number of outer and inner steps M and N and the flow head layers H. It shows a general trend of achieving better performance with a larger effective NFE and TMD offers a more fine-grained control over this performance-efficiency tradeoff than DMD2-*v*.

B.2. Curvature of Wan trajectories

Inspired by Liu et al. [34], we define the curvature of Wan's sampling trajectory at time t as

$$C(\boldsymbol{v}, t) = \|\boldsymbol{v}(\boldsymbol{x}_t, t) - (\boldsymbol{x} - \boldsymbol{x}_0)\|^2.$$

Since the model is evaluated on a fixed time grid during sampling, we adopt a discretized version of the curvature in practice, *i.e.*,

$$C(\boldsymbol{v}, t_i) = \left\| \frac{\tilde{\boldsymbol{x}}_{t_i} - \tilde{\boldsymbol{x}}_{t_{i-1}}}{t_i - t_{i-1}} - (\boldsymbol{x}_1 - \boldsymbol{x}) \right\|^2,$$

where t_i denotes the timesteps from Wan's default 50-step sampling schedule and $\tilde{\boldsymbol{x}}_{t_i}$ represents the corresponding intermediate samples along the trajectory.

As shown in Figure 12, the sampling trajectories of the Wan model exhibit extremely large curvature near $t = 1$ (*i.e.*, the high noise regime), making it difficult for trajectory matching methods to learn the mapping along the ODE path. This also motivates our choice of larger γ in the time-shifting function for r_i in TM-MF (see Section A.2) and t_{student} in DMD2-*v* (see Section 3.2).

B.3. Flow head conditioning

Our main experiments use the gating mechanism explained in Section A.1 to condition the flow head on the main backbone's features $\boldsymbol{m} = \boldsymbol{m}(\boldsymbol{x}, t)$. Another version is to concatenate $\boldsymbol{m}$ and the patch embeddings of $\boldsymbol{y}_s$ along the hidden dimension and then project to the original hidden size using a linear layer. To minimize the impact on the pretrained model, we initialized the weights of the linear layer as identity for the coordinates corresponding to $\boldsymbol{m}$ and as a Gaussian with small standard deviation (0.01) for the remaining coordinates. In Table 9, we see that TMD can achieve strong performance across fusion types. While convergence was more stable using the gating mechanism used for our main experiments (see Figure 13), concatenation is a strong alternative in terms of final performance.

Figure 1	
Short prompt	*A fat rabbit wearing a purple robe walking through a fantasy landscape.*
Long prompt	*A plump, fluffy rabbit donning a voluminous purple robe walks gracefully through a vibrant fantasy landscape. The rabbit has large, expressive eyes and a gentle, curious expression. Its fur is soft and thick, and the robe drapes elegantly over its body. The landscape features rolling hills covered in lush green grass, colorful wildflowers, and towering magical trees with shimmering leaves. In the distance, there are sparkling waterfalls and mystical castles. The scene is bathed in warm, golden sunlight. Medium shot, focusing on the rabbit's walk through the picturesque environment.*
Short prompt	*A person drinking coffee in a cafe.*
Long prompt	*A cozy, warm café setting with soft ambient lighting and wooden furnishings. A young adult, casually dressed in a sweater and jeans, sits at a small round table. They hold a steaming cup of coffee in their hand, taking a sip while looking pensively out the window. The café is moderately busy with other patrons engaged in conversations. The background showcases various coffee drinks and pastries displayed on a counter. The person's expression is relaxed and content. Medium shot focusing on the person's face and the coffee cup, capturing the intimate atmosphere of the café.*
Figure 3	
Short prompt	*A person is laughing.*
Long prompt	*A joyful person is laughing heartily, with a broad smile and crinkled eyes, conveying pure happiness. They are standing upright with arms spread wide, as if embracing the world around them. The scene is set outdoors in a sunny park, surrounded by lush greenery and blooming flowers. The background includes a clear blue sky with fluffy clouds, adding to the cheerful atmosphere. Medium shot capturing the full body of the person, focusing on their animated facial expressions and gestures.*
Short prompt	*A steam train moving on a mountainside.*
Long prompt	*A vintage steam locomotive chugging along a winding track on a mountainous terrain. The train is covered in soot, with steam billowing from its smokestack as it navigates the rugged landscape. The surrounding mountains are steep and lush, with patches of snow visible at higher elevations. The train cars sway gently as they follow the curving tracks, and the scenery outside the windows shows dense forests and rocky cliffs. The camera follows the train from a medium distance, capturing the train's movement and the dramatic backdrop.*
Figure 4	
Short prompt	*A boat sailing leisurely along the Seine River with the Eiffel Tower in background.*
Long prompt	*A serene, picturesque scene of a small wooden boat gently gliding along the Seine River in Paris, France. The boat is rowed leisurely by a middle-aged man in a casual striped shirt and khaki pants, who rows smoothly with rhythmic strokes. The Eiffel Tower stands majestically in the background, partially visible through the misty morning air. The riverbank is lined with lush green trees and quaint buildings, reflecting off the calm waters. The overall atmosphere is peaceful and tranquil, capturing the essence of a lazy summer day. Wide shot, static camera.*
Short prompt	*An astronaut flying in space, zoom out.*
Long prompt	*Astronaut floating in space with a helmet visor reflecting Earth below. The astronaut, wearing a full spacesuit with the American flag on the shoulder, is performing a spacewalk, arms extended as if in motion. The background shows the vastness of space with stars twinkling and Earth in the distance. The scene begins with a close-up of the astronaut and gradually zooms out to reveal the enormity of space surrounding them. Wide shot, showcasing the astronaut against the backdrop of the universe.*
Negative prompt	
Bright tones, overexposed, static, blurred details, subtitles, style, works, paintings, images, static, overall gray, worst quality, low quality, JPEG compression residue, ugly, incomplete, extra fingers, poorly drawn hands, poorly drawn faces, deformed, disfigured, misshapen limbs, fused fingers, still picture, messy background, three legs, many people in the background, walking backwards	

Table 8. Extended prompts used in the figures from the main text and negative prompt used for CFG (in teacher sampling, KD, and DMD2-*v*) and condition dropout (in TM-MF), taken from the official Wan repository.

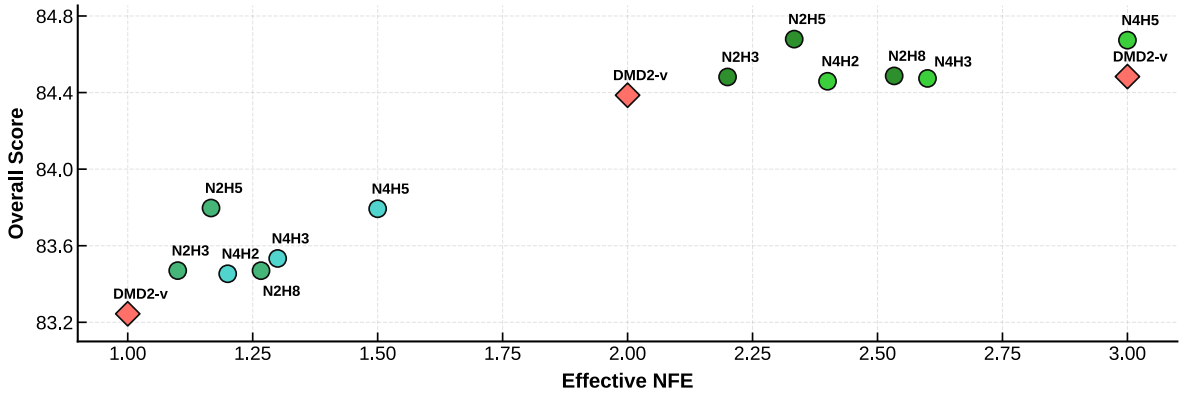


Figure 11. **Performance-efficiency tradeoff of TMD.** Extension of Figure 6 to include the $M = 1$ settings.

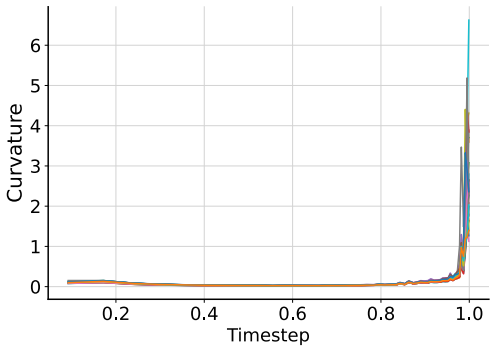


Figure 12. **Curvature of Wan trajectories.** Curvature of different sampling trajectories for the Wan2.1 1.3B model. Large trajectory curvatures are observed near $t = 1$ (*i.e.*, the high noise regime).

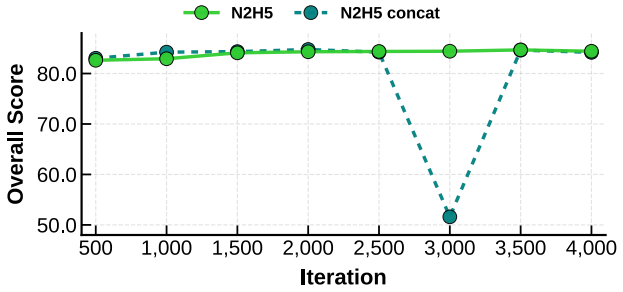


Figure 13. **Convergence and fusion ablation.** We compare the overall VBench score over iterations for the second-stage TMD training with our gating mechanism (see Section A.1) and concatenation of features (see Section B.3). While concatenation yields very competitive results, we observed that it introduces some training instabilities.

Setting	Fusion type	Overall score	Quality score	Semantic score
N2H5	Gated (see Section A.1)	84.68	85.71	80.55
	Concat	**84.76**	**85.77**	**80.71**
N4H5	Gated (see Section A.1)	**84.67**	85.72	**80.47**
	Concat	84.66	**85.79**	80.14

Table 9. Impact of the fusion type on the final performance when distilling Wan2.1 1.3B with $M = 2$.

B.4. Inner flow targets

In principle, the auxiliary latent variable $\boldsymbol{y}$ in the inner flow can be chosen arbitrarily as long as $\boldsymbol{x}_{t_{i-1}}$ is easy to sample given $\boldsymbol{y}$ and $\boldsymbol{x}_{t_i}$. While we chose the DTM formulation $\boldsymbol{y} := \boldsymbol{x}_1 - \boldsymbol{x}$ in (3) for our main experiments, we also experimented with other versions. While other formulations did not improve the overall performance, they could still achieve competitive performance. For instance, Table 10 provides results for the simple choice $\boldsymbol{y} := \boldsymbol{x}$. Similar to (14) and (17), we parametrize the average velocity as

$$\boldsymbol{u}_\theta(\boldsymbol{y}_s, s, r; \boldsymbol{m}) := \frac{\boldsymbol{y}_s - (\boldsymbol{x}_{t_i} - t_i \mathrm{head}_\theta(\boldsymbol{y}_s, s, r; \boldsymbol{m}))}{s}.$$

where $\boldsymbol{m} = \boldsymbol{m}_\theta(x_{t_i}, t_i)$ denote the main features.

B.5. Impact of recurrence in flow head

To evaluate the effect of recurrence in the flow head, we manually restrict the inner flow to a single step at inference (N1H5) and compare the results with the standard recurrent setting (N4H5). As shown in Figure 14, when distilling Wan2.1 1.3B into a two-step generator, the non-recurrent variant produces noticeably lower-quality videos, exhibiting stronger artifacts and blurriness. This highlights the importance of iterative refinement within the flow head for high-fidelity generation.

Setting	Target $\boldsymbol{y}$	Overall score	Quality score	Semantic score
N2H5	$\boldsymbol{x}_1 - \boldsymbol{x}$ (DTM)	**84.68**	**85.71**	**80.55**
	$\boldsymbol{x}$	84.18	85.15	80.33
N4H5	$\boldsymbol{x}_1 - \boldsymbol{x}$ (DTM)	**84.67**	**85.72**	80.47
	$\boldsymbol{x}$	84.44	85.36	**80.77**

Table 10. Impact of the inner flow target when distilling Wan2.1 1.3B with $M = 2$.

B.6. DMD2 comparisons

In Section 3.2, we observed that KD pretaining is only helpful for one-step distillation (see Table 4 and Figure 9) and that timestep shifting of t_{dmd} and t_{student} is crucial for good performance for one- and two-step DMD2-*v* (see Table 5 and Figure 10). In this section, we provide additional ablations on our DMD2-*v* baseline when using ≥ 3 steps. While the timestep shifting for t_{dmd} and t_{student} remain important, we observed that slightly lower shift values for t_{student} can attain better scores when using more steps. In particular, for 4 steps, we picked a shift value of $\gamma = 5$ for t_{student}, outperforming the DMD2 baseline in [69]; see Table 11.

Method	shift t_{student}	NFE	Overall score	Quality score	Semantic score
DMD2 [69]		4	84.56	85.58	**80.50**
DMD2-*v*	10	4	84.53	85.95	78.84
DMD2-*v*	5	4	**84.60**	**86.03**	79.87
DMD2-*v*	10	3	**84.48**	**85.71**	79.58
DMD2-*v*	5	3	84.47	85.55	**80.15**

Table 11. VBench results of the DMD2 version in [69] and our DMD2-*v* for 3 and 4 steps with different timestep shifts for t_{student}.

B.7. Inference time

In Table Table 12, we measure the inference time of our proposed decoupled architecture to confirm that the flow head

is lightweight compared to the backbone. Moreover, the gray columns validate the usage of *effective* NFE for evaluating efficiency, since the increase in the per-step inference time closely matches the effective NFE.

Setting	Backbone	Head	Fusion	Total	Increase	NFE
N4H5 (1.3B)	783	156	4.8	1426	1.51x	1.5
N2H5 (1.3B)	781	155	4.7	1100	1.17x	1.17
N4H5 (14B)	4191	595	29	6687	1.4x	1.38

Table 12. Inference time (in milliseconds) of different architecture components with $M = 1$ and batch-size 1 on a single H100 GPU. The increase in inference time (second-to-last column) is measured relative to a standard forward pass of the teacher model and closely matches the effective NFE (last column).

B.8. Training iterations

We note that TMD is training-efficient compared to other baselines. For instance, for Wan2.1 14B and $M = 2$ we only require 3k iterations of lightweight TM-MF pretraining and 1k iterations of distribution matching, whereas rCM uses 10k iterations (with VSD and sCM losses) [69]. For Wan2.1 1.3B, we show in Table 13 that even with significantly reduced number of TM-MF iterations, TMD still outperforms rCM and DMD2-*v*.

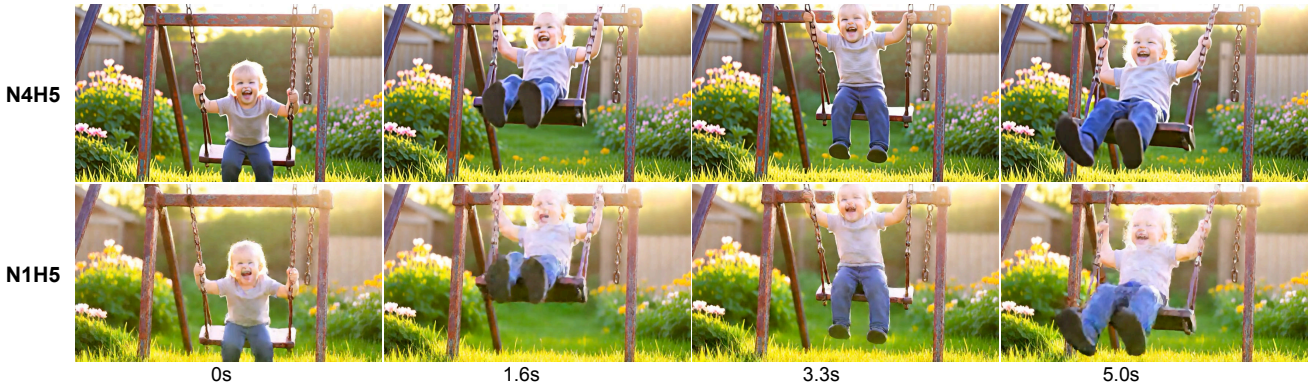
Method	TM-MF (iters)	Distill (iters)	Overall score	Quality score	Semantic score
rCM [69]	-	10k	84.09	84.90	**80.86**
DMD2-*v*	-	<6k	84.39	<u>85.65</u>	79.32
TMD-N2H5	0.5k	<6k	84.52	<u>85.65</u>	79.99
TMD-N2H5	1k	<6k	<u>84.59</u>	85.57	<u>80.68</u>
TMD-N2H5	3k	<6k	**84.68**	**85.71**	80.55

Table 13. Comparison of VBench scores for distilling Wan2.1 1.3B using TMD ($M = 2$) with a varying number of TM-MF pretraining iterations against two-step baselines.

B.9. More visual comparison results

We provide further visual comparisons between the 50-step teacher models with classifier-free guidance (CFG), DMD2-*v*, and TMD in Figures 15 to 20.

A joyful child, with a big smile and arms spread wide, swings energetically on a rusty old swing set in a sunlit backyard. The swing set, with peeling paint and creaking chains, contrasts against the vibrant green grass and blooming flowers surrounding it. The child's laughter echoes as they swing higher and higher, their feet barely touching the ground at the bottom of each arc.



Impressionist style, a single yellow rubber duck gently floating on undulating waves at sunset. The duck has a cheerful smile painted on its face, with sunlight casting warm, golden hues across the water. The sky is filled with soft, pastel shades of orange, pink, and purple, reflecting off the rippling surface. The water appears calm yet lively, with subtle waves carrying the duck along smoothly.

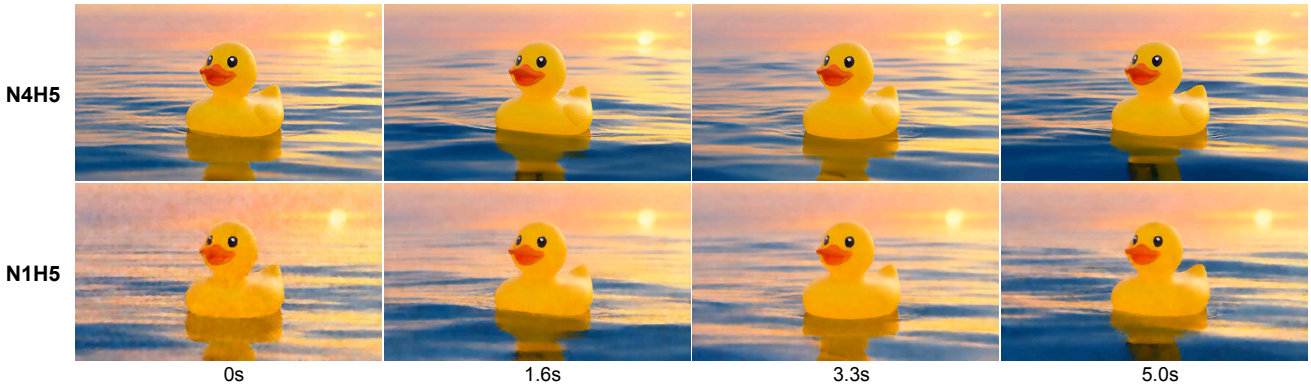


Figure 14. **Impact of flow head recurrence.** We show the impact of recurrence in the flow head by setting the number of flow head steps to 1 only at inference (*i.e.*, N1H5) when distilling Wan2.1 1.3B with $M = 2$ and the N4H5 setting for flow head (*i.e.*, 4 denoising steps and 5 DiT blocks in flow head). We observe that the videos generated without recurrence (marked by N1H5) are of much lower quality (*e.g.*, more artifacts and blurriness) than ones with recurrence (marked by N4H5), implying the importance of the fine-grained iterative refinement on our method.

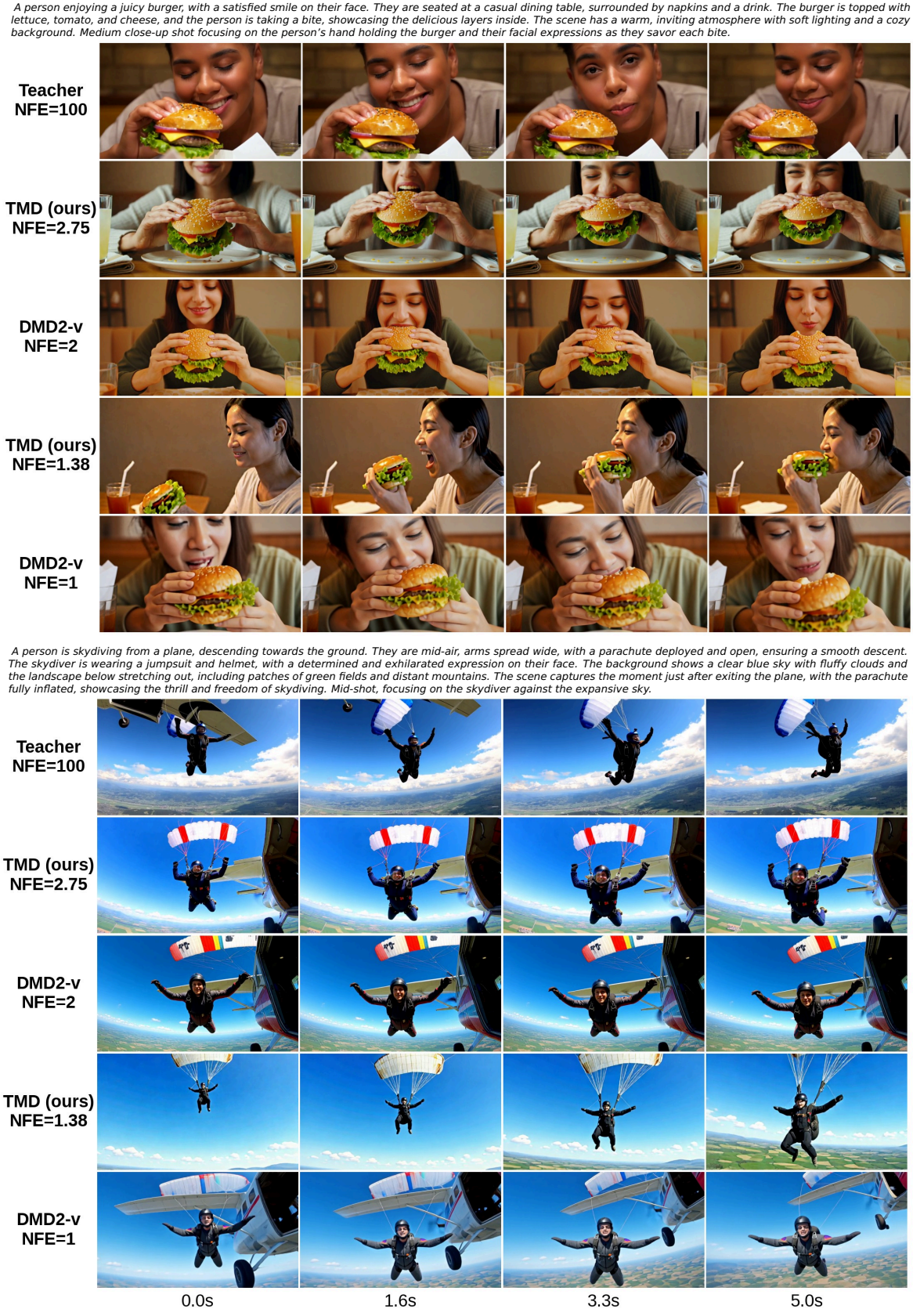


Figure 15. **Visual comparison on Wan2.1 14B.** We compare the outputs of the teacher, TMD, and DMD2-*v* on exemplary prompts.

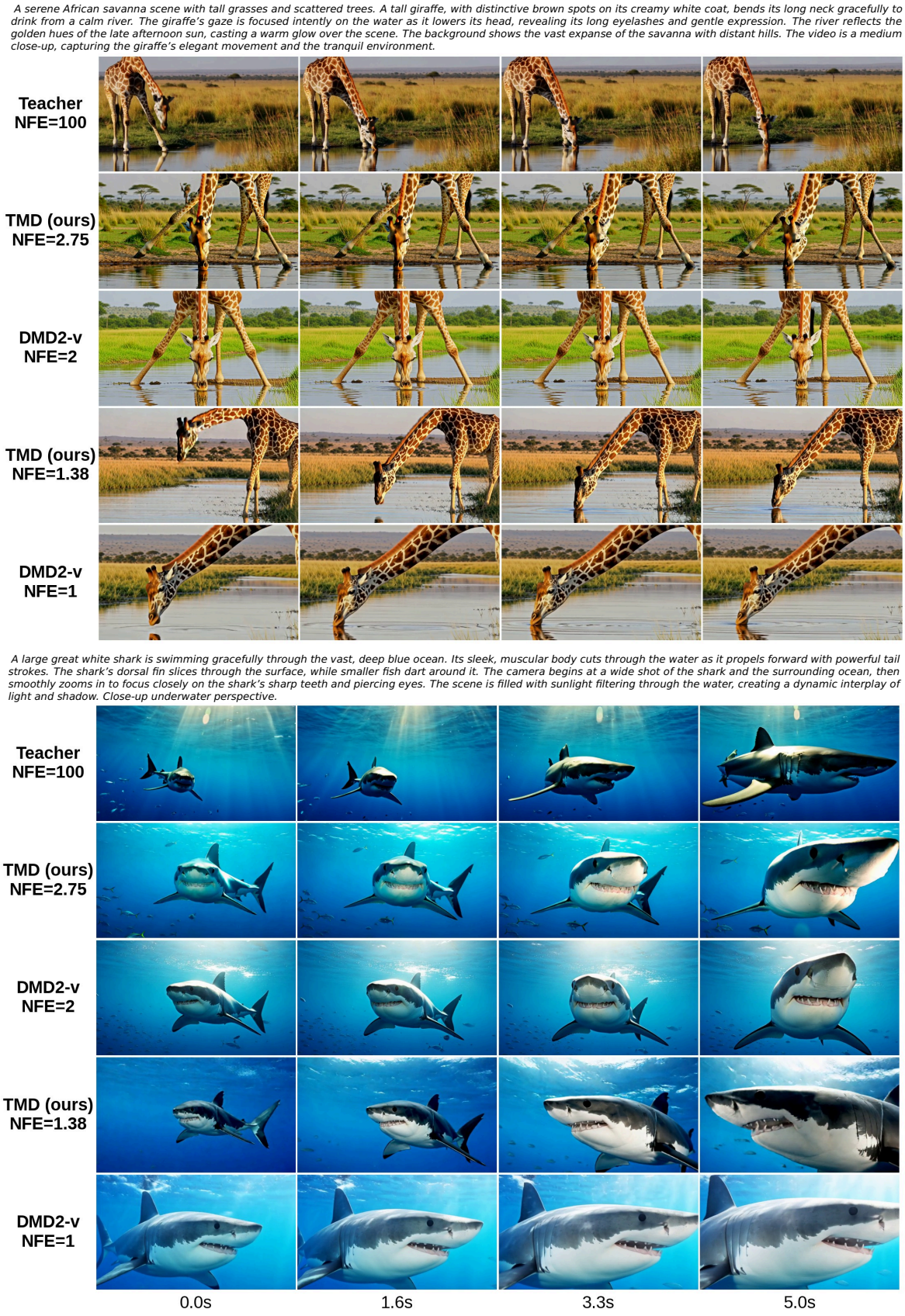


Figure 16. **Visual comparison on Wan2.1 14B.** We compare the outputs of the teacher, TMD, and DMD2-*v* on exemplary prompts.

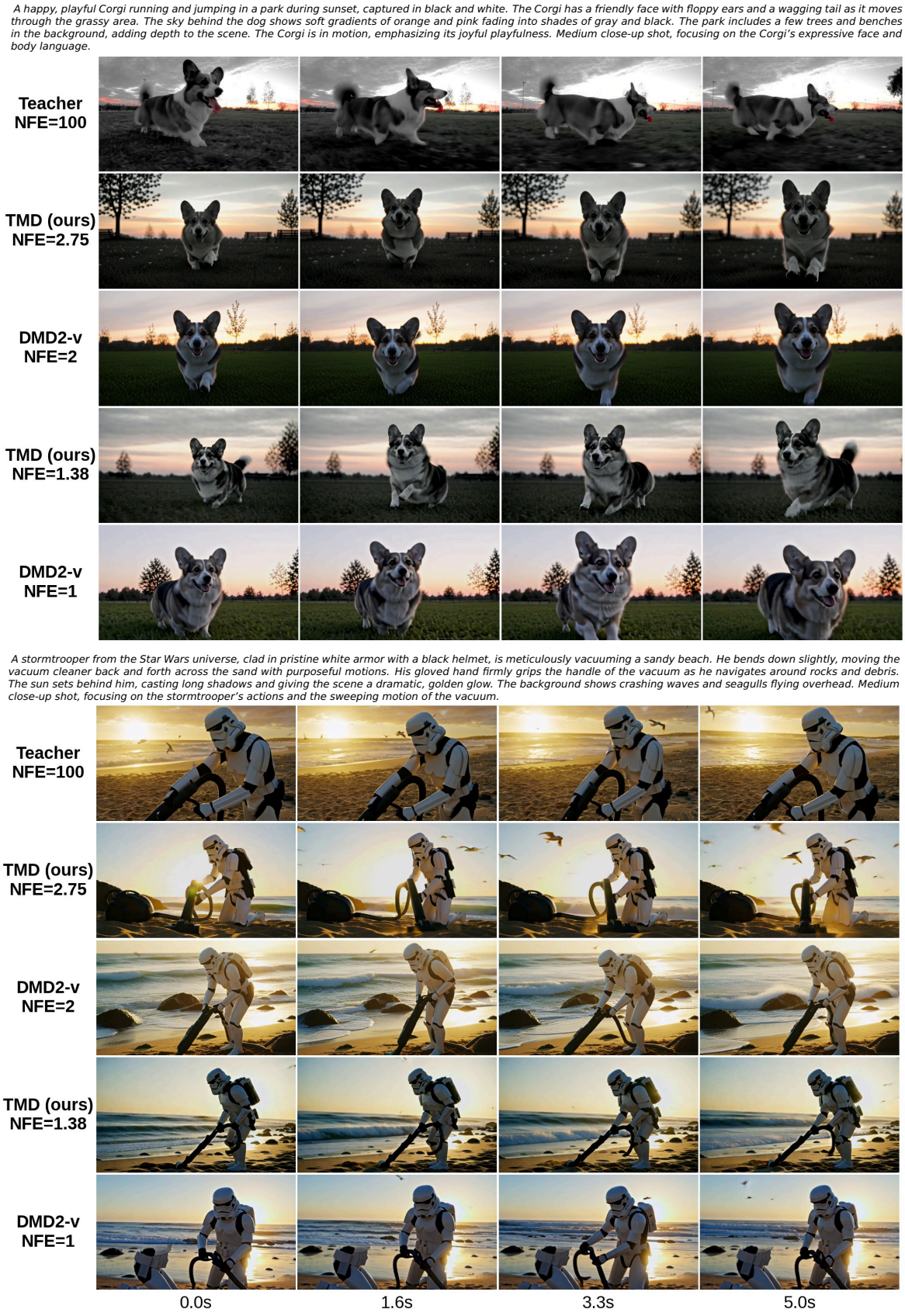


Figure 17. **Visual comparison on Wan2.1 14B.** We compare the outputs of the teacher, TMD, and DMD2-*v* on exemplary prompts.

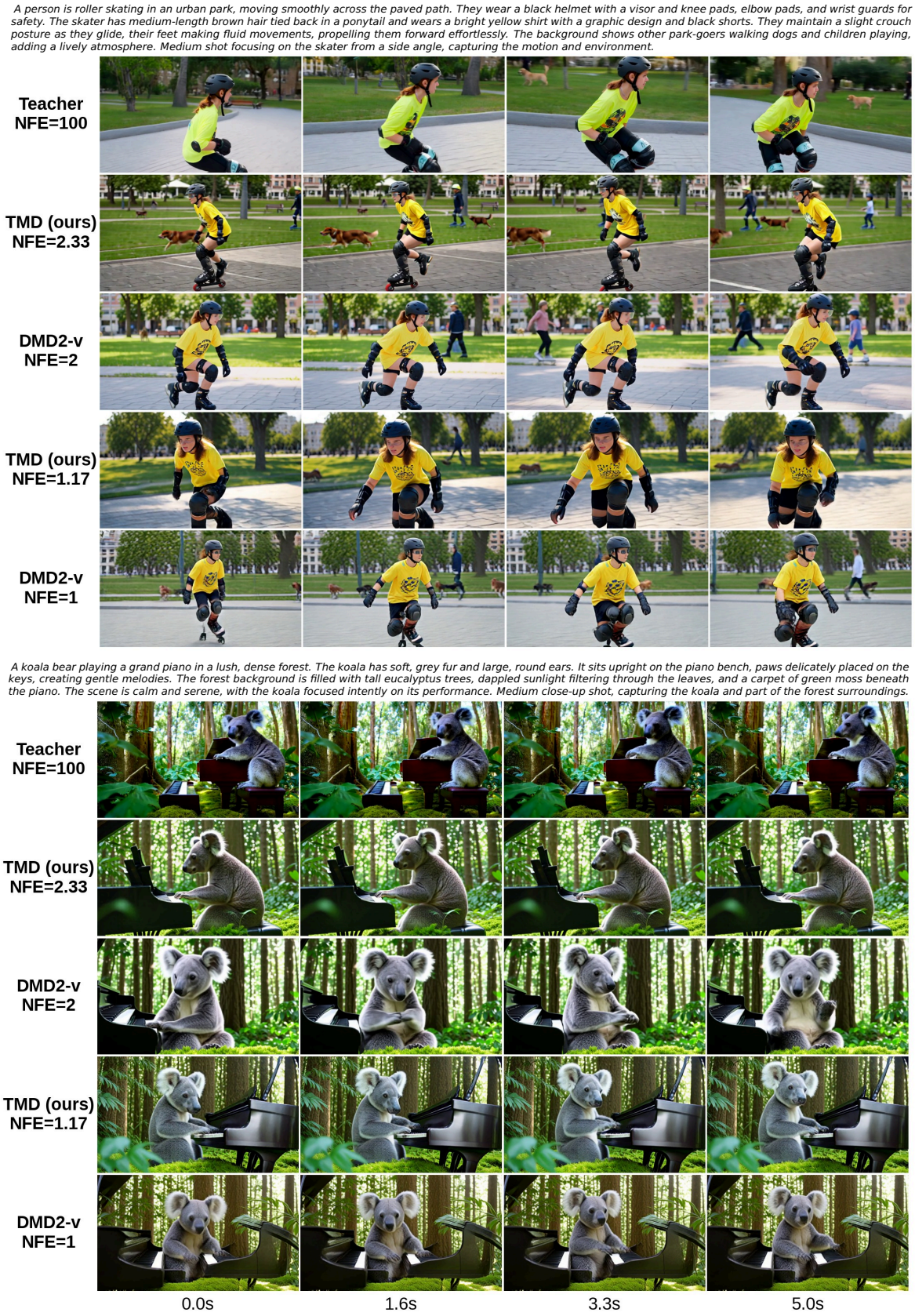


Figure 18. **Visual comparison on Wan2.1 1.3B.** We compare the outputs of the teacher, TMD, and DMD2-*v* on exemplary prompts.

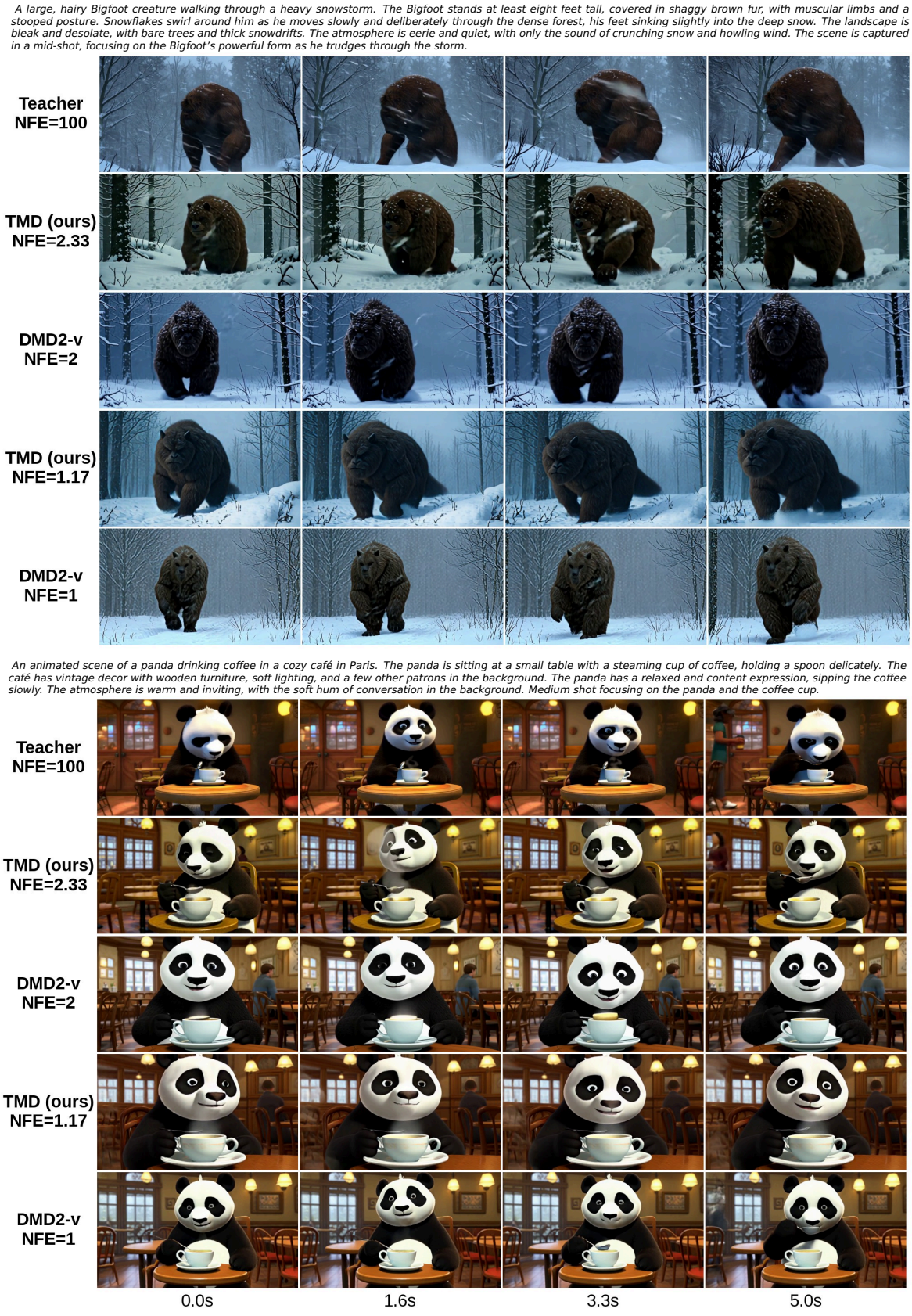


Figure 19. **Visual comparison on Wan2.1 1.3B.** We compare the outputs of the teacher, TMD, and DMD2-*v* on exemplary prompts.

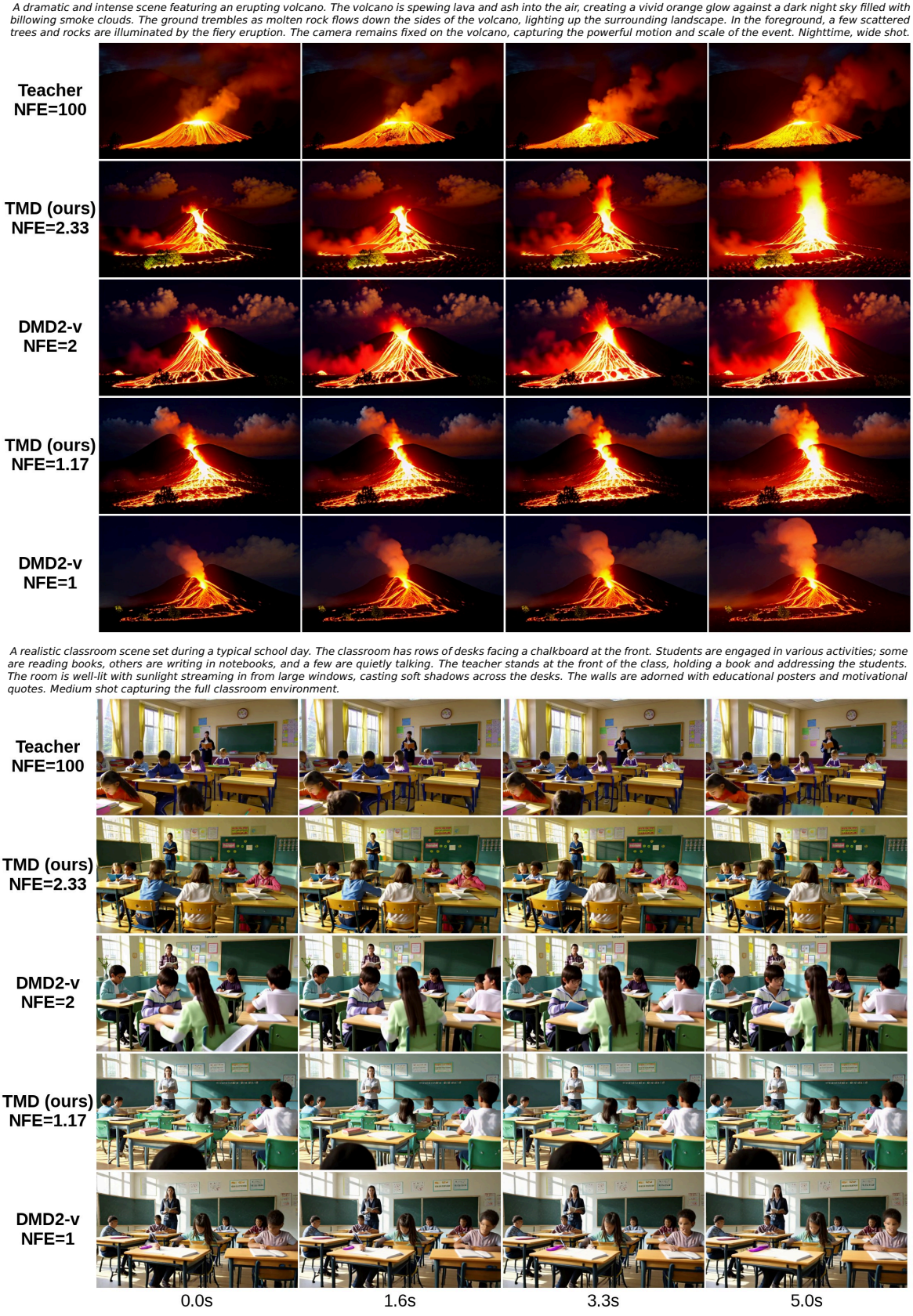


Figure 20. **Visual comparison on Wan2.1 1.3B.** We compare the outputs of the teacher, TMD, and DMD2-*v* on exemplary prompts.

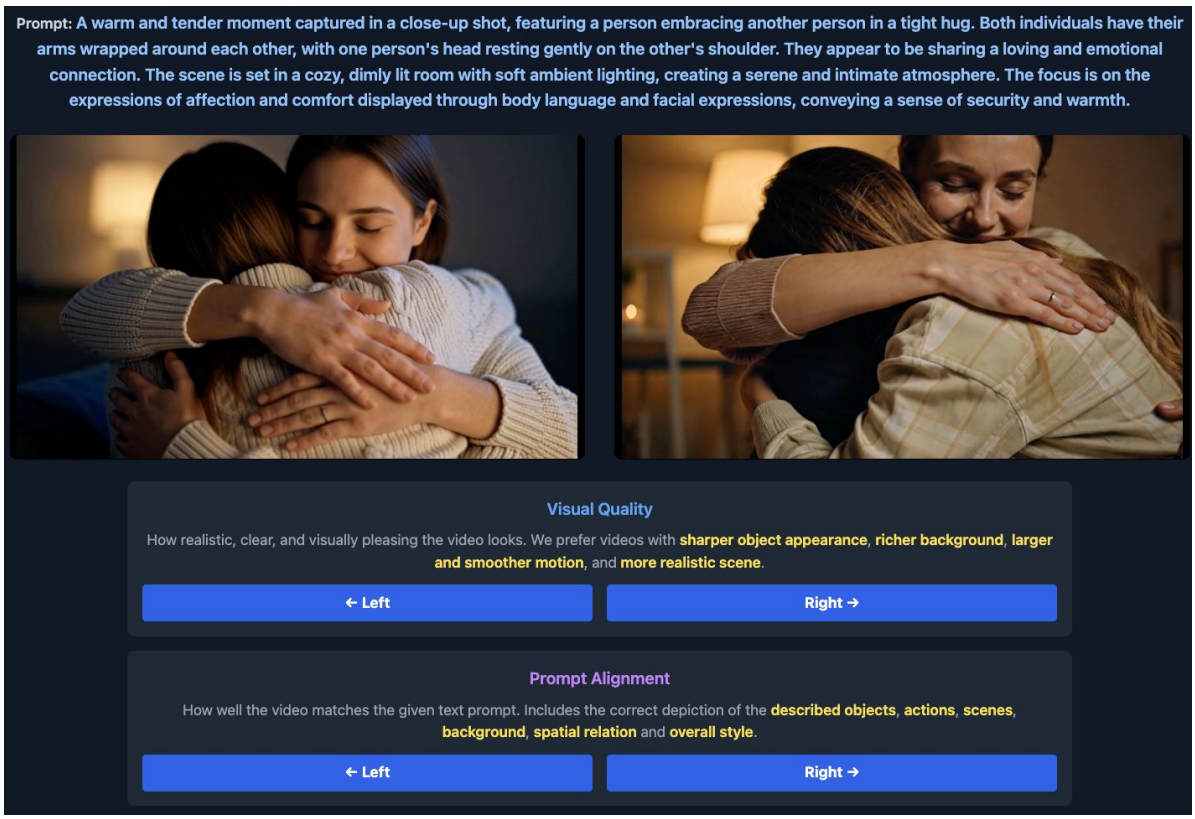


Figure 21. **User study interface.** Screenshot of our user preference study interface explained in Section 4.2.