RoPECraft: Training-Free Motion Transfer with Trajectory-Guided RoPE Optimization on Diffusion Transformers

Ahmet Berke Gokmen*, Yigit Ekin*, Bahri Batuhan Bilecen*, Aysegul Dundar

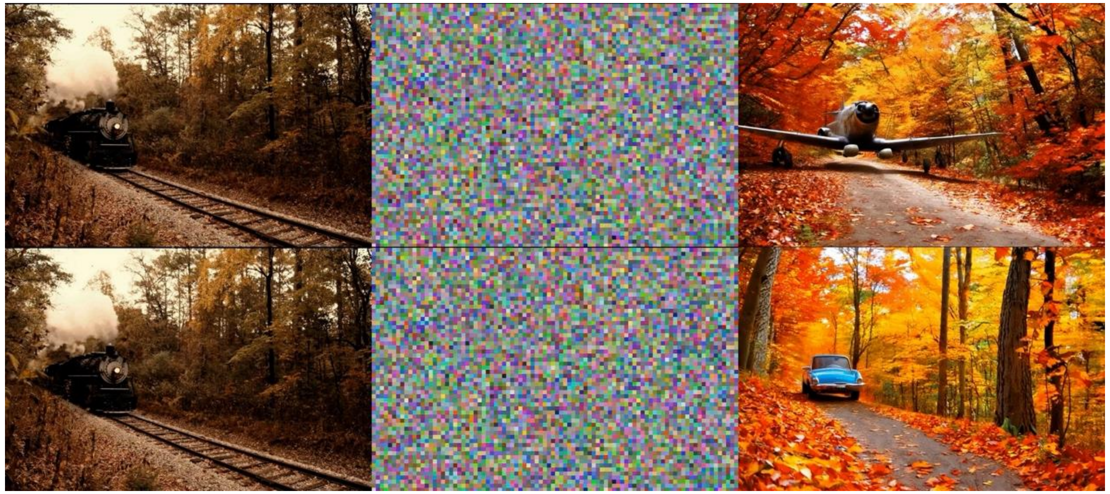
*: equal contribution

Noise warping

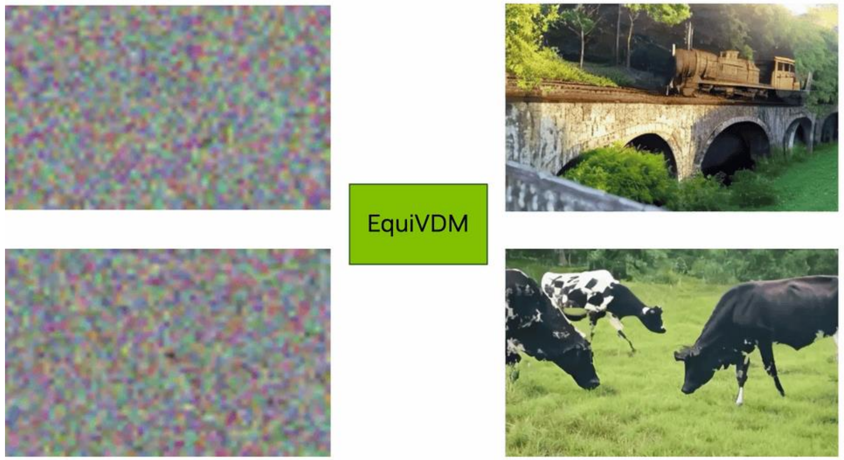
How I Warped Your Noise
ICLR 2024 Oral, ETHZ

Leverages generative **image diffusion models** for consistent video generation via noise warping

Adapting to video models



Go-with-the-Flow, CVPR 2025 Oral, Netflix



EquiVDM, 2025, NVIDIA

Fine-tune the model to understand warped noise behavior
Used for controllable generation in general

A problem with noise warping

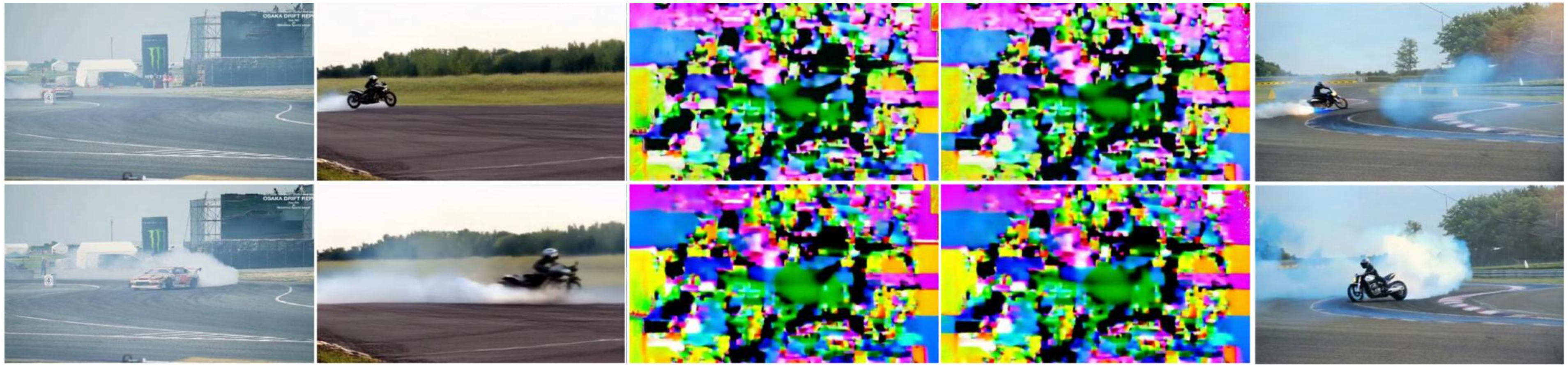
- Noise warping changes the latent space. This means the network needs fine-tuning & re-training. For GWTF[1], takes 40-GPU days of fine-tuning on pre-trained CogVideoX!
- If we utilize the same noise warping algorithm on other backbones (like Wan), it collapses
- Test-time optimization does not recover the sample, either

Input reference | *GWTF [1] (trained on CogVideoX with 128-dim LoRAs)* | *GWTF's noise warping on Wan 2.1* | *GWTF's noise warping on Wan 2.1 + test-time optimization*

[1] Burgert et al., Go-with-the-Flow, CVPR 2025

Our approach: RoPE warping

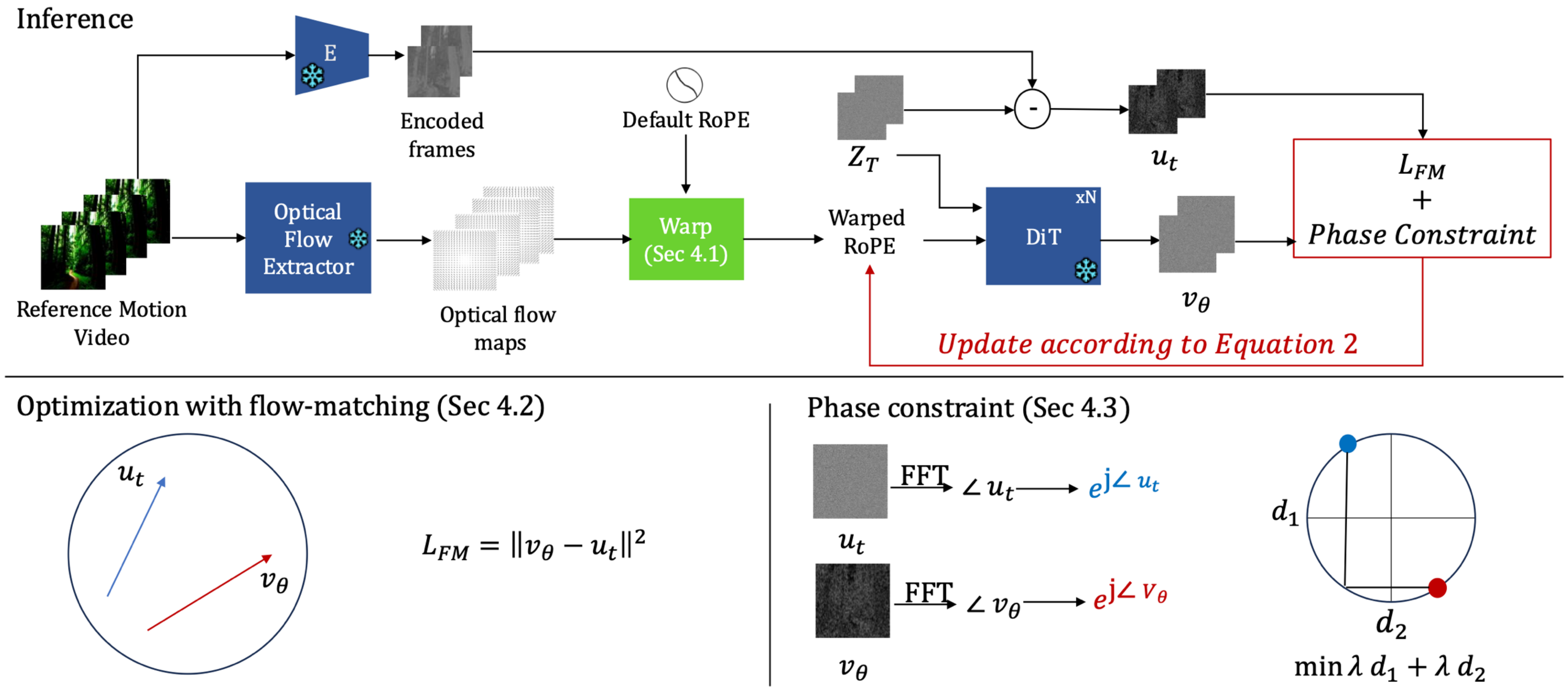
- Rather than noise warping, **we explore rotary positional embedding (RoPE) warping on motion transfer tasks** [RoPECraft, NeurIPS 2025]
- This does not require any training and can work on any video diffusion backbone!
- We are the first ones exploring RoPE warping on video models. Similartly, later works also explored video super-resolution [2], novel-view synthesis [3], camera embeddings [4], etc.



Input reference	*GWTF [1] (trained on CogVideoX with 128-dim LoRAs)*	*GWTF's noise warping on Wan 2.1*	*GWTF's noise warping on Wan 2.1 + test-time optimization*	***Our approach, no training***

[2] *Issachar et al., DyPE: Dynamic Position Extrapolation for Ultra High-Resolution Diffusion, 2025*
[3] *Bai et al., Positional Encoding Field, 2025*
[4] *Li et al., Camera as Relative Positional Encoding, 2025*

Pipeline


Inference
E
Encoded frames
Default RoPE
Z_T
-
u_t
L_{FM} + Phase Constraint
Reference Motion Video
Optical Flow Extractor
Optical flow maps
Warp (Sec 4.1)
Warped RoPE
xN
DiT
v_θ
Update according to Equation 2
Optimization with flow-matching (Sec 4.2)
u_t
v_θ
$L_{FM} = \|v_\theta - u_t\|^2$
Phase constraint (Sec 4.3)
FFT
$\angle u_t$
$e^{j\angle u_t}$
u_t
FFT
$\angle v_\theta$
$e^{j\angle v_\theta}$
v_θ
d_1
d_2
$\min \lambda d_1 + \lambda d_2$

RoPE in video models

$$< q\Phi_a, k\Phi_b > = Re\left(qk^T \cdot e^{j(a-b)f}\right) = qk^T \boldsymbol{\cos((a-b)f)}$$

In attention maps qk^T, the value is **scaled** by the relative position between $(a-b)$

Algorithm 1 Default 1D RoPE, expanded to 3D

Input: Base frequency $\theta \in \mathbb{R}_{>0}$
Embedding dims $D_t, D_h, D_w \in \mathbb{N}$
Sequence lengths $S_t, S_h, S_w \in \mathbb{N}$
for each $k \in \{t, h, w\}$ **do**
  $\mathbf{p} = [0, 1, \ldots, S_k - 1]^{\mathrm{T}}$ $\quad \triangleright \in \mathbb{R}^{S_k}$
  $\mathbf{d} = [0, 1, \ldots, D_k/2 - 1]^{\mathrm{T}}$ $\quad \triangleright \in \mathbb{R}^{D_k/2}$
  $\mathbf{f} = \theta^{-2\mathbf{d}/D_k}$ $\quad \triangleright \in \mathbb{R}^{D_k/2}$
  $\mathbf{\Phi}_k = e^{j\mathbf{p}\mathbf{f}^{\mathrm{T}}}$ $\quad \triangleright \in \mathbb{C}^{S_k \times (D_k/2)}$
  $\mathbf{\Phi}_k = \texttt{expand}(\mathbf{\Phi}_k)$ $\quad \triangleright \in \mathbb{C}^{S_t \times S_h \times S_w \times (D_k/2)}$
end for
$\mathbf{\Phi} = \texttt{concat}(\mathbf{\Phi}_{t,h,w})$ $\quad \triangleright \in \mathbb{C}^{S_t \times S_h \times S_w \times (D/2)}$
$\mathbf{\Phi} = \texttt{flatten}(\mathbf{\Phi})$ $\quad \triangleright \in \mathbb{C}^{1 \times 1 \times (S_t S_h S_w) \times (D/2)}$

Embeddings are calculated for (t,h,w) dimensions for 1D, and expanded (broadcasted) for the other 2 dims

This means as t changes, we still have the same (h,w) embeddings.

The network understands this (because trained with it), but we can also manipulate it to our advantage to encourage a controllable generation

1) RoPE warping

Algorithm 1 Default 1D RoPE, expanded to 3D

Input: Base frequency $\theta \in \mathbb{R}_{>0}$
Embedding dims $D_t, D_h, D_w \in \mathbb{N}$
Sequence lengths $S_t, S_h, S_w \in \mathbb{N}$
for each $k \in \{t, h, w\}$ **do**
  $\mathbf{p} = [0, 1, \dots, S_k - 1]^{\mathrm{T}}$ $\triangleright \in \mathbb{R}^{S_k}$
  $\mathbf{d} = [0, 1, \dots, D_k/2 - 1]^{\mathrm{T}}$ $\triangleright \in \mathbb{R}^{D_k/2}$
  $\mathbf{f} = \theta^{-2\mathbf{d}/D_k}$ $\triangleright \in \mathbb{R}^{D_k/2}$
  $\mathbf{\Phi}_k = e^{j\mathbf{p}\mathbf{f}^{\mathrm{T}}}$ $\triangleright \in \mathbb{C}^{S_k \times (D_k/2)}$
  $\mathbf{\Phi}_k = \texttt{expand}(\mathbf{\Phi}_k)$ $\triangleright \in \mathbb{C}^{S_t \times S_h \times S_w \times (D_k/2)}$
end for
$\mathbf{\Phi} = \texttt{concat}(\mathbf{\Phi}_{t,h,w})$ $\triangleright \in \mathbb{C}^{S_t \times S_h \times S_w \times (D/2)}$
$\mathbf{\Phi} = \texttt{flatten}(\mathbf{\Phi})$ $\triangleright \in \mathbb{C}^{1 \times 1 \times (S_t S_h S_w) \times (D/2)}$

Algorithm 2 Motion-augmented RoPE

Input: Base frequency $\theta \in \mathbb{R}_{>0}$,
Embedding dims $D_t, D_h, D_w \in \mathbb{N}$
Sequence lengths $S_t, S_h, S_w \in \mathbb{N}$
Optical flows $\mathbf{u}, \mathbf{v}$ $\triangleright \in \mathbb{R}^{2 \times S_t \times H \times W}$

$\mathbf{u}, \mathbf{v} = \texttt{downsample}(\mathbf{u}, \mathbf{v})$ $\triangleright \in \mathbb{R}^{2 \times S_t \times S_h \times S_w}$
$\mathbf{h}_{\text{flow}}, \mathbf{w}_{\text{flow}} = \texttt{cumsum}(\mathbf{u}, \mathbf{v})$
$\mathbf{h}_{\text{flow}} = \texttt{flatten}(\mathbf{h}_{\text{flow}})$ $\triangleright \in \mathbb{R}^{(S_t \times S_w) \times S_h}$
$\mathbf{w}_{\text{flow}} = \texttt{flatten}(\mathbf{w}_{\text{flow}})$ $\triangleright \in \mathbb{R}^{(S_t \times S_h) \times S_w}$

$\mathbf{f}_h = \theta^{-2[0,1,\dots,D_h/2-1]^{\mathrm{T}}/D_h}$ $\triangleright \in \mathbb{R}^{D_h/2}$
for each r in $[0, 1, \dots, S_t \times S_w]$ **do**
  $\mathbf{p} = [0, 1, \dots, S_h - 1]^{\mathrm{T}} + \boxed{\mathbf{h}_{\text{flow}}[r]}$ $\triangleright \in \mathbb{R}^{S_h}$
  $\mathbf{\Phi}_h[r] = e^{j\mathbf{p}\mathbf{f}_h^{\mathrm{T}}}$ $\triangleright \in \mathbb{C}^{S_h \times (D_h/2)}$
end for
$\mathbf{\Phi}_h = \texttt{reorder}(\mathbf{\Phi}_h)$ $\triangleright \in \mathbb{C}^{S_t \times S_h \times S_w \times (D_h/2)}$

$\mathbf{f}_w = \theta^{-2[0,1,\dots,D_w/2-1]^{\mathrm{T}}/D_w}$
for each c in $[0, 1, \dots, S_t \times S_h]$ **do**
  $\mathbf{p} = [0, 1, \dots, S_w - 1]^{\mathrm{T}} + \boxed{\mathbf{w}_{\text{flow}}[c]}$
  $\mathbf{\Phi}_w[c] = e^{j\mathbf{p}\mathbf{f}_w^{\mathrm{T}}}$
end for
$\mathbf{\Phi}_w = \texttt{reorder}(\mathbf{\Phi}_w)$

$\mathbf{p} = [0, \dots, S_t - 1]^{\mathrm{T}}$
$\mathbf{f} = \theta^{-2[0,\dots,D_t/2-1]^{\mathrm{T}}/D_t}$
$\mathbf{\Phi}_t = \texttt{expand}(e^{j\mathbf{p}\mathbf{f}^{\mathrm{T}}})$

$\mathbf{\Phi} = \texttt{flatten}(\texttt{concat}(\mathbf{\Phi}_{t,h,w}))$

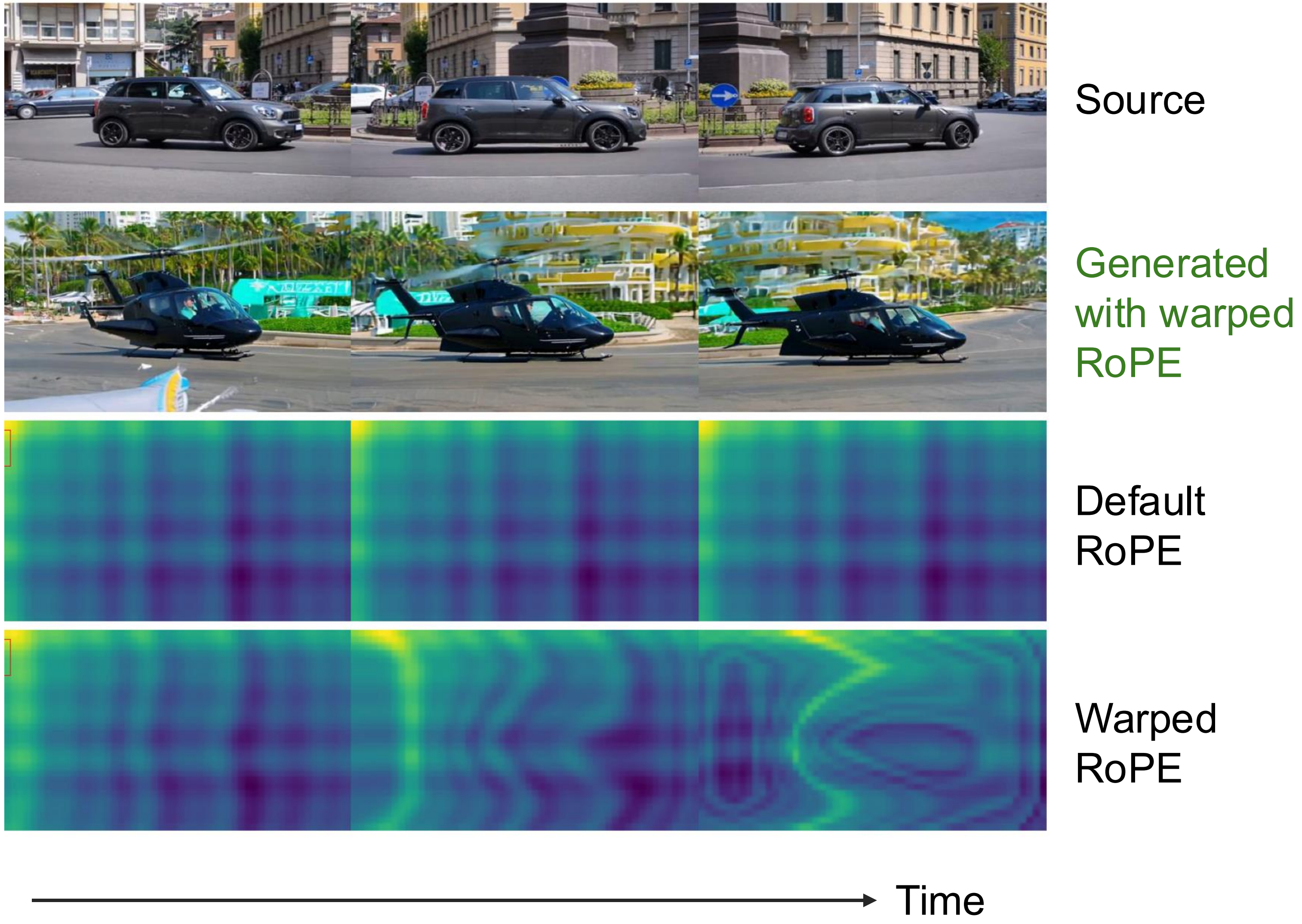
Let us not broadcast to (h,w) dimensions, but offset them based on the optical flow cues

For instance, the network behaves as if the upper-left indexed patch is changing its location, as time goes on

1) RoPE warping

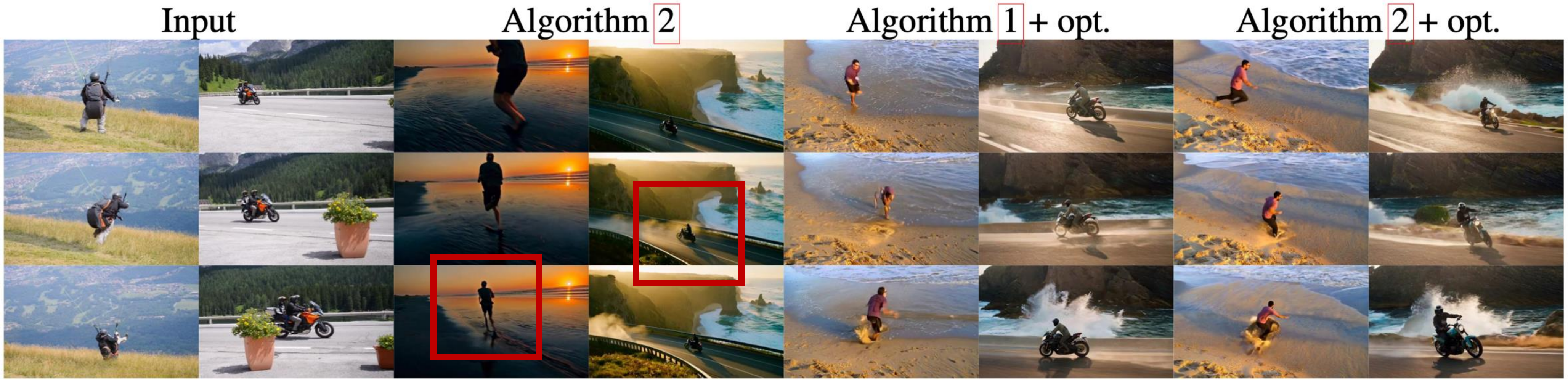
Visualization of warped RoPE

Prompt: A helicopter taking off



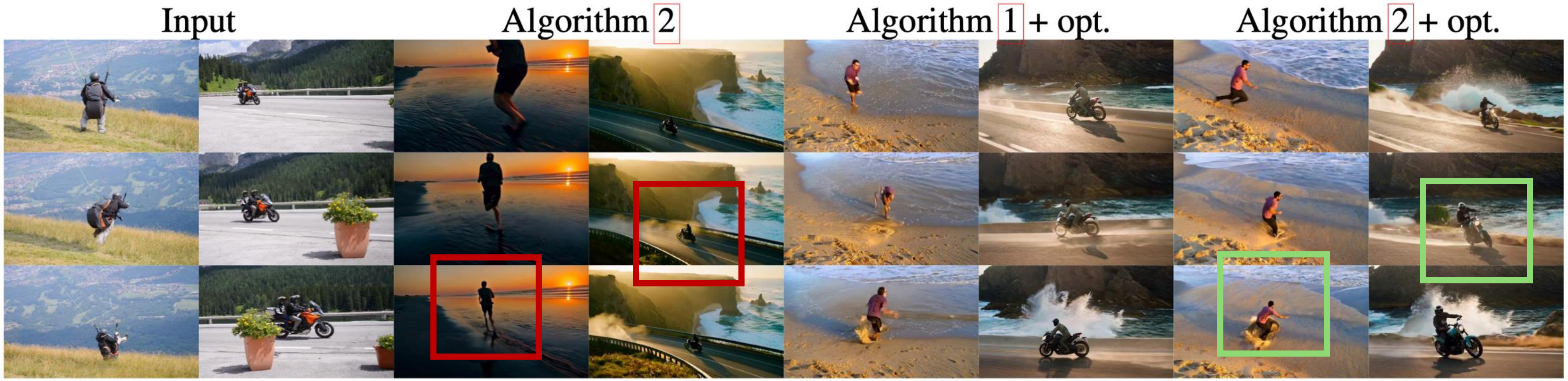
2) Per-sample optimization on top of warping

- Even though RoPE warping is sufficient, in some cases, subjects may be oriented in the opposite way



2) Per-sample optimization on top of warping

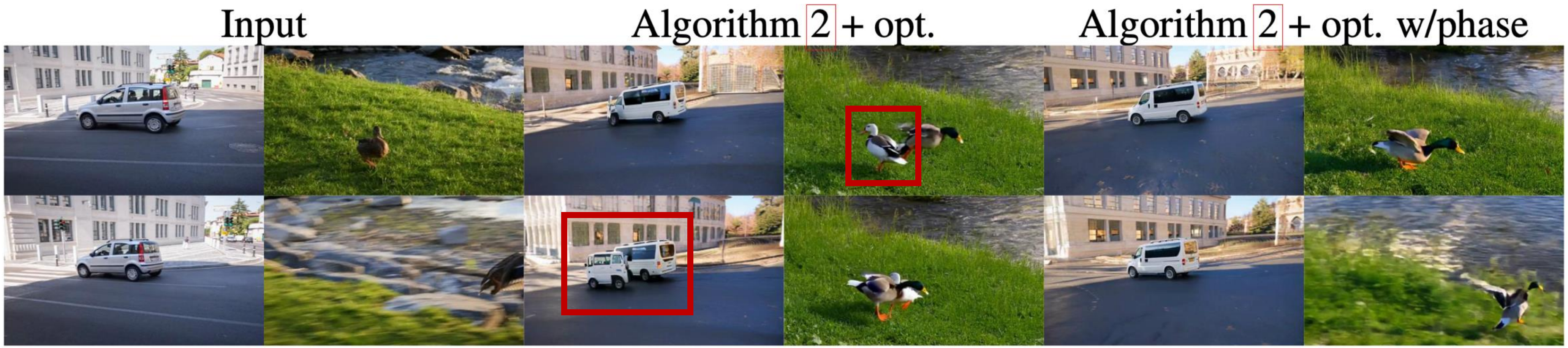
- Even though RoPE warping is sufficient, in some cases, subjects may be oriented in the opposite way



- To refine it, we align the generated velocity in early generation steps (cos-sim) $v_\theta(t, x_t)$, with the target velocity $u_t = \sigma_t^{-1}(x_t - \boldsymbol{v})$, where:
  - v_θ is the DiT output
  - x_t is the input latent at timestep t
  - $\boldsymbol{v}$ is the clear latent reference video
  - σ_t is the schedular sigma at timestep t

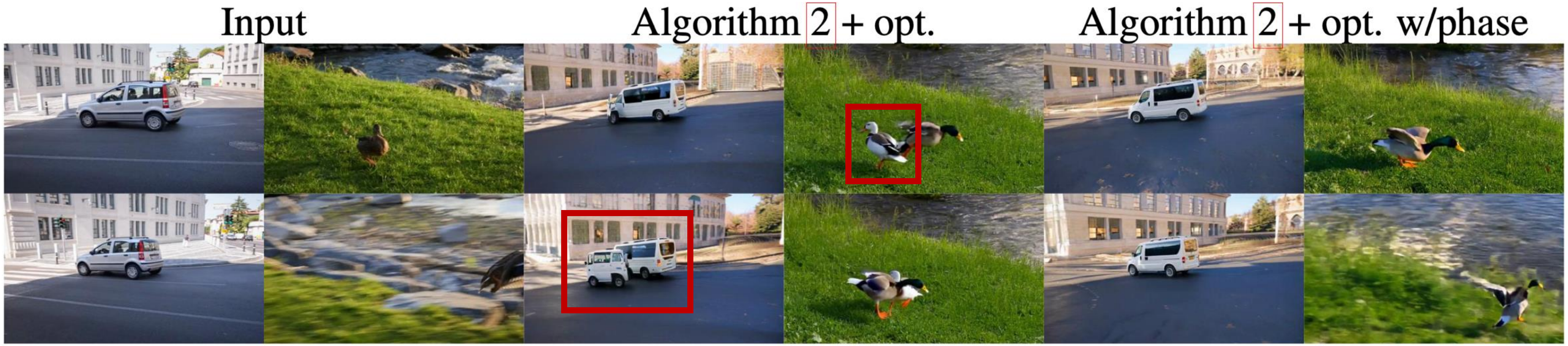
$$Loss = \mathcal{L}_{FM}(u_t, v_\theta) = cossim(u_t, v_\theta)$$

3) Phase constraints



- After RoPE warping, some RoPE tensors may be aliased (meaning that not in the correct frequency range), which may cause feature copying

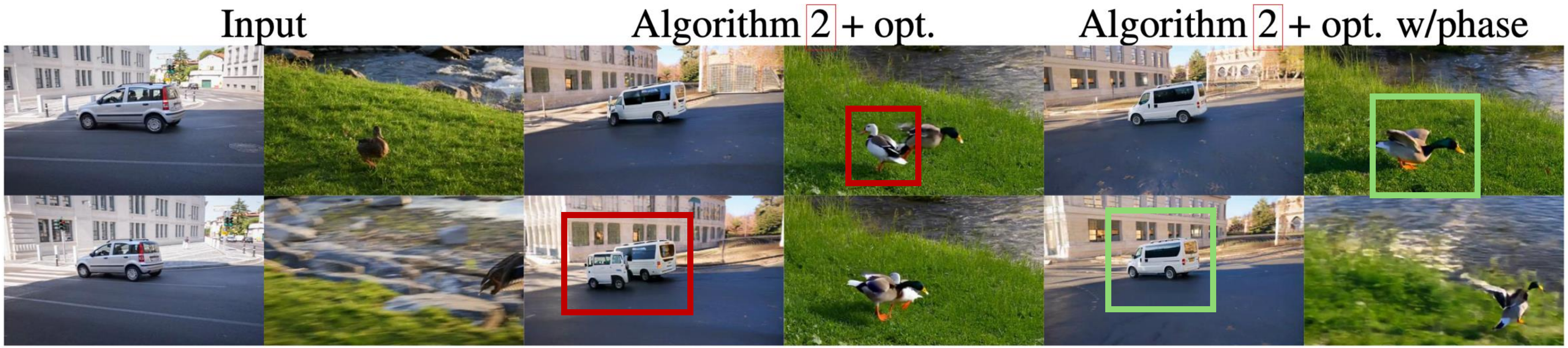
3) Phase constraints



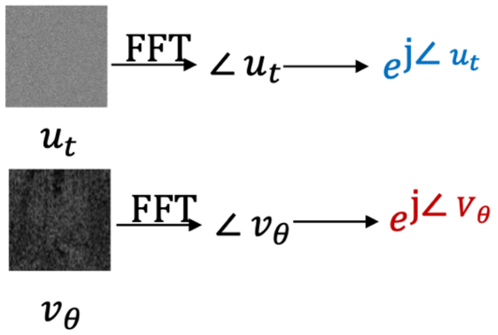
- After RoPE warping, some RoPE tensors may be aliased (meaning that not in the correct frequency range), which may cause feature copying
- We got inspired by Fourier domain properties, namely the shift (a shift in spatial domain is identical to a phase scaling)

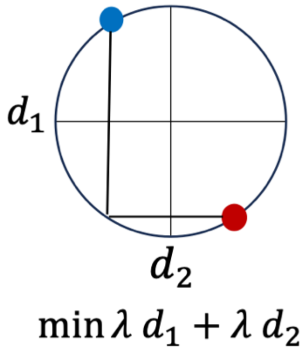
$$f(x-a, y-b) \Leftrightarrow e^{j2\pi(au+bv)} F(u,v)$$

3) Phase constraints



- Therefore, we also added a Fourier phase constraint between generated velocity and the target velocity, to eliminate the unwanted copying:





$$Loss = \mathcal{L}_{FM}(u_t, v_\theta) + \lambda \| \cos\angle F(u_t) - \cos\angle F(v_\theta) \| + \lambda \| \sin\angle F(u_t) - \sin\angle F(v_\theta) \|$$

Fréchet Trajectory Distance

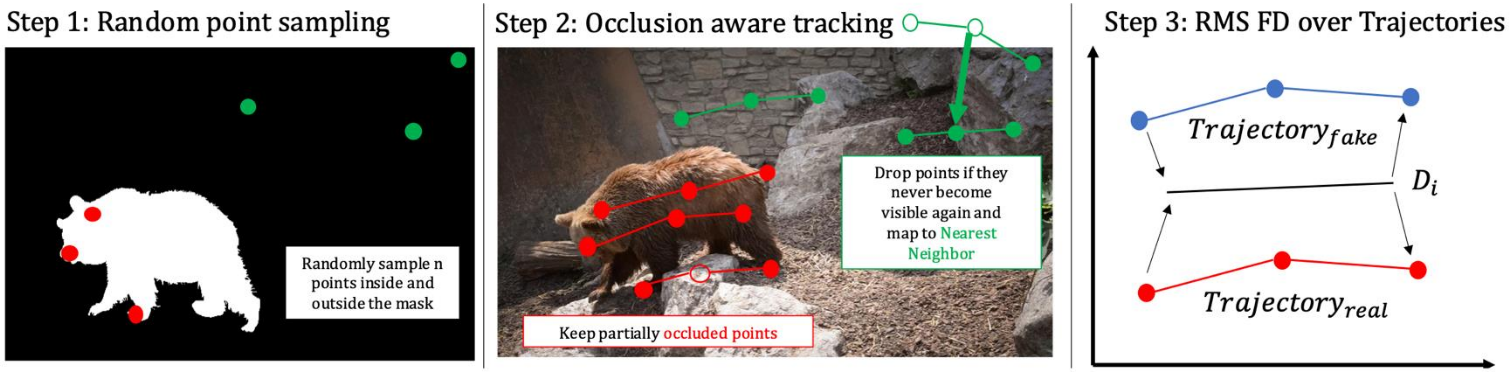


Figure 8: **Fréchet Trajectory Distance (FTD). 1)** Sample n foreground (**red**) and n background (**green**) seeds on the first frame. **2)** Track each seed with an occlusion-aware filler: copy the nearest visible neighbor while occluded and discard tracks that never re-appear. **3)** Measure the RMS Fréchet distance between generated (**fake**) and reference (**real**) tracks.

- We also propose Fréchet Trajectory Distance, where we assign random trajectories and compute the Fréchet distance between them
- We show that it is more reliable than Motion Fidelity [Yatim et al., CVPR 2024] for motion transfer tasks

$$D_F\left(\mathcal{T}_i^{\text{real}}, \mathcal{T}_i^{\text{fake}}\right) = \min_{\sigma,\tau:\{1,\ldots,L\}\twoheadrightarrow\{1,\ldots,T\}} \max_{k=1,\ldots,L} \left\|\mathbf{x}_{i,\sigma(k)}^{\text{real}} - \mathbf{x}_{i,\tau(k)}^{\text{fake}}\right\|_2,$$

$$\text{FTD} = \left(N^{-1}\Sigma_{i=1}^{N} D_F^2(\mathcal{T}_i^{\text{real}}, \mathcal{T}_i^{\text{fake}})\right)^{0.5}$$

Quantitative results

Table 1: Comparison of motion transfer methods across evaluation metrics. Best and second results are represented with *italic* and underlined, respectively.

Method	MF ↑	CD-FVD ↓	CLIP ↑	FTD (FG) ↓	FTD (FG+BG) ↓
GWTF [4]	0.5713±0.22	1485.23	*0.2378±0.04*	*0.2457±0.14*	*0.2308±0.10*
SMM [45]	0.4889±0.20	1600.33	0.2331±0.04	0.2882±0.15	0.3176±0.15
MOFT [42]	0.4606±0.20	1630.45	0.2311±0.04	0.2811±0.16	0.3057±0.14
DitFlow (latents) [31]	0.4832±0.20	1735.49	0.2339±0.04	0.2921±0.15	0.3135±0.12
DitFlow (RoPE) [31]	0.4500±0.18	1852.90	0.2345±0.04	0.2785±0.14	0.3019±0.13
ConMo [14]	0.4627±0.21	1680.78	0.2309±0.04	0.2769±0.15	0.3040±0.14
Ours	*0.5816±0.19*	*1284.58*	0.2350±0.04	0.2644±0.14	0.2584±0.13

Table 2: Ablation on motion-augmented RoPE and phase constraints.

Method	MF	CLIP	FTD
Alg.1 + opt.	0.7082	0.1560	0.2174
Alg.2 + opt.	0.7092	0.1650	0.2105
Alg.2 + opt. + phase	*0.7210*	*0.1656*	*0.2060*

Table 3: Ablation on hyperparameters.

(t, s)	MF	CD-FVD	CLIP	FTD
$5, 5$	0.5165	1437.99	0.1597	0.2901
$5, 10$	0.5523	1606.88	0.1663	0.2728
$10, 5$	0.5675	*1364.25*	*0.1664*	0.2633
$10, 10$	*0.6160*	1492.86	0.1572	*0.2573*

Qualitative results

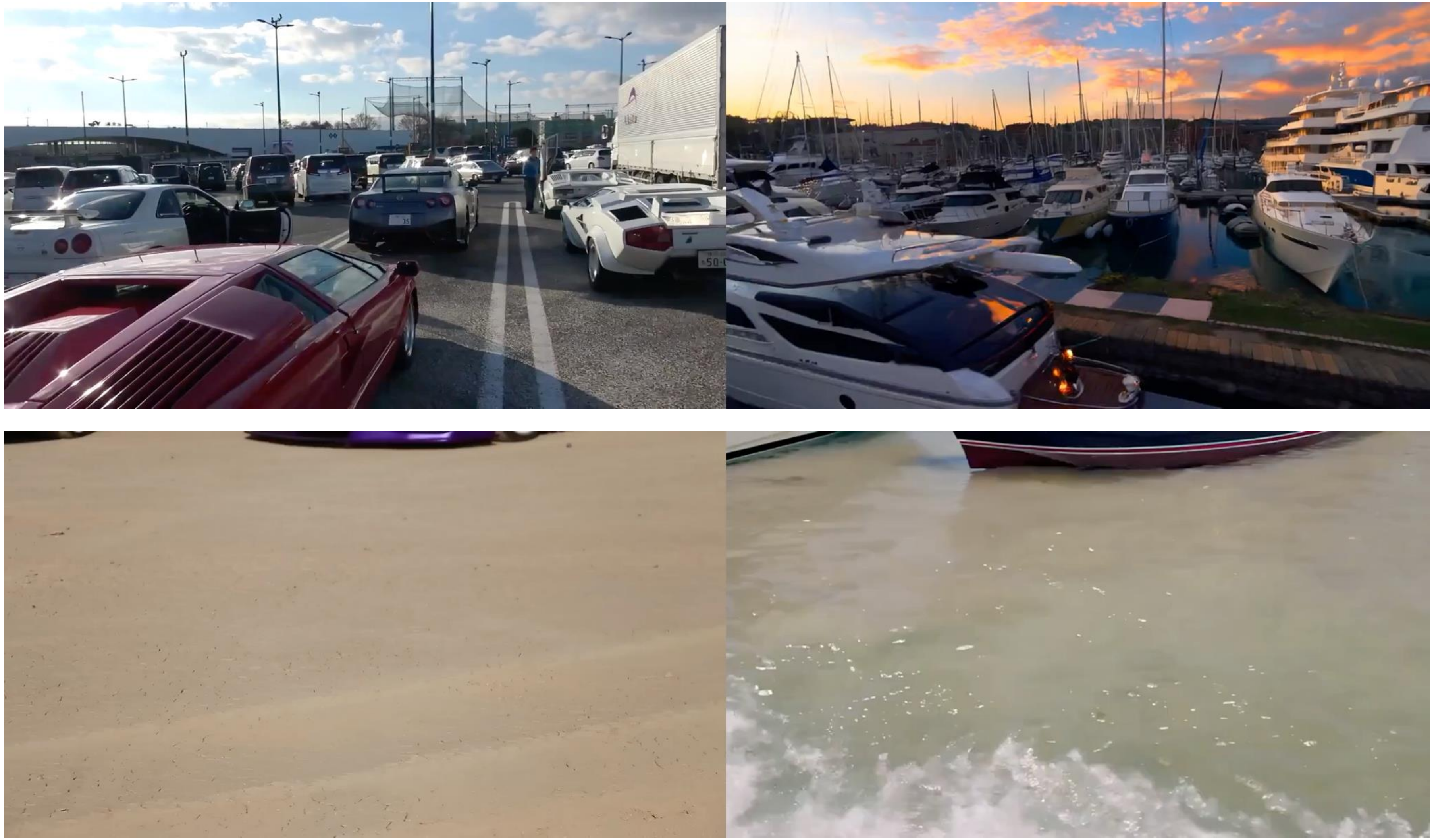
- Noise warping is also too strict in terms of obeying the object shape, and cannot adapt to the text prompt well
- We developed our method on Wan2.1-1.3B T2V and one-shot tested on CogVideoX-5B T2V:

Input reference | ***Ours (RoPECraft), on CogVideoX*** | *GWTF on CogVideoX* | *DitFlow on CogVideoX*

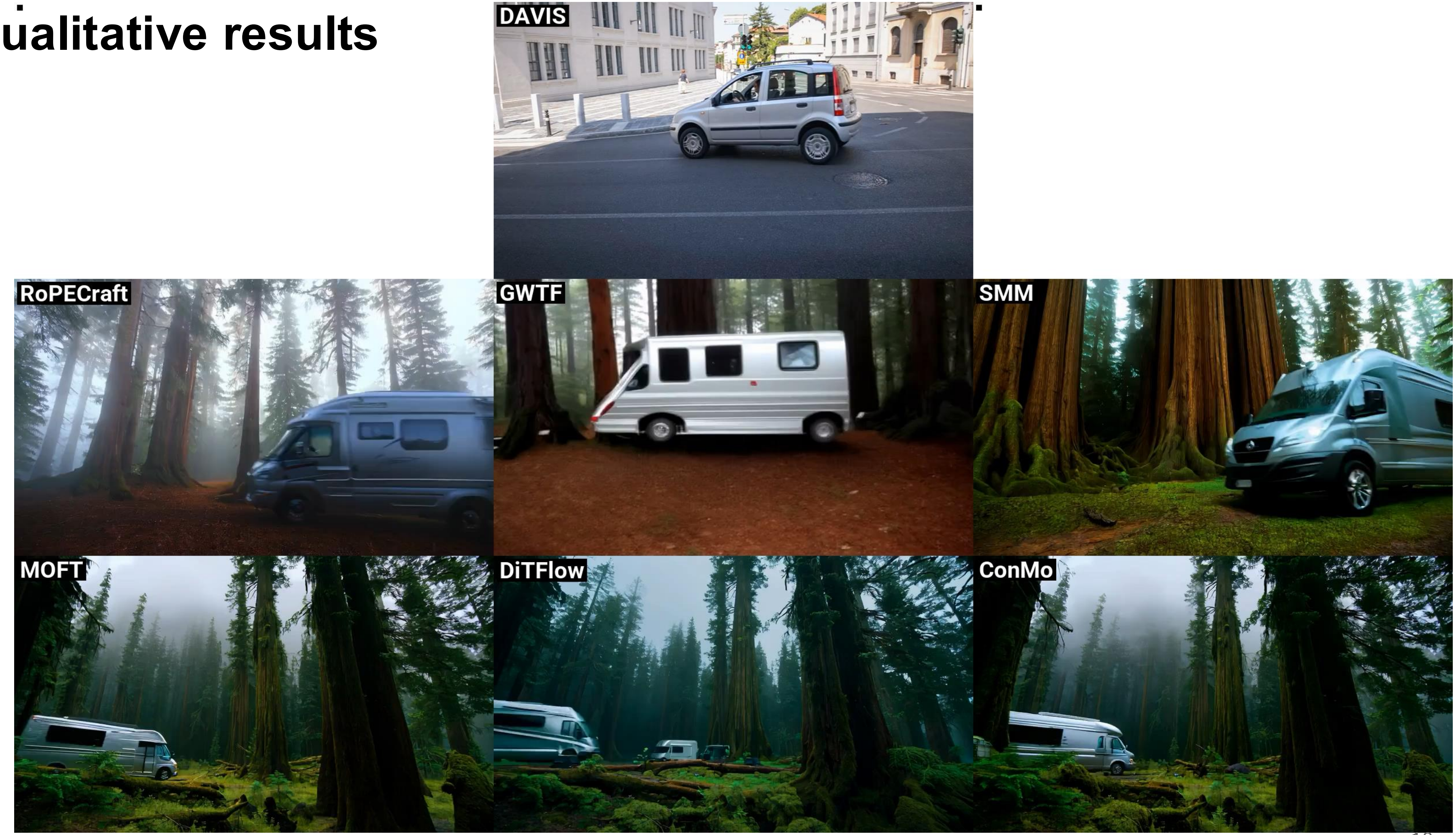
Qualitative results

- RoPECraft can also handle camera control in video generation:

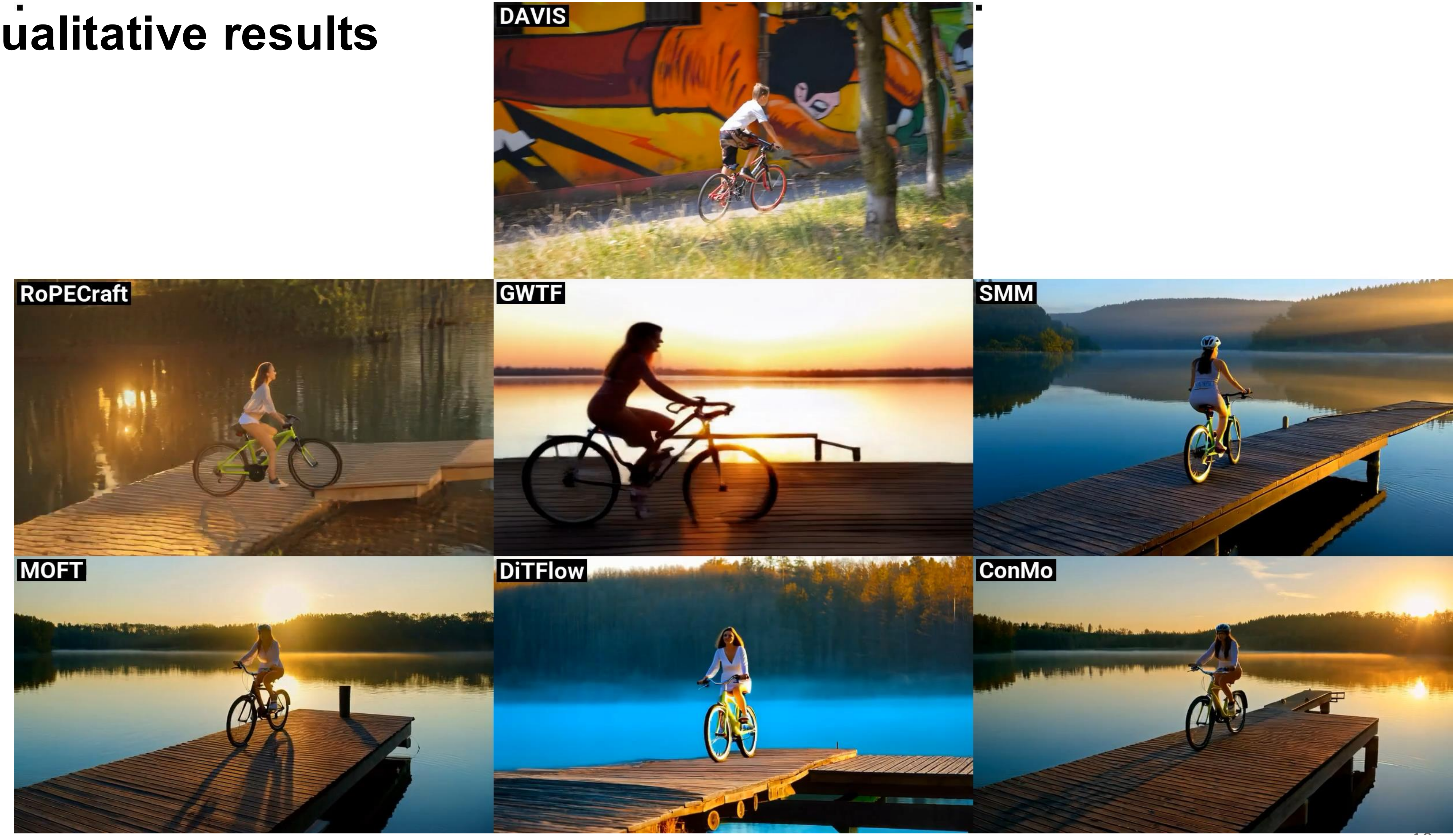
Input reference **_Ours (RoPECraft)_**



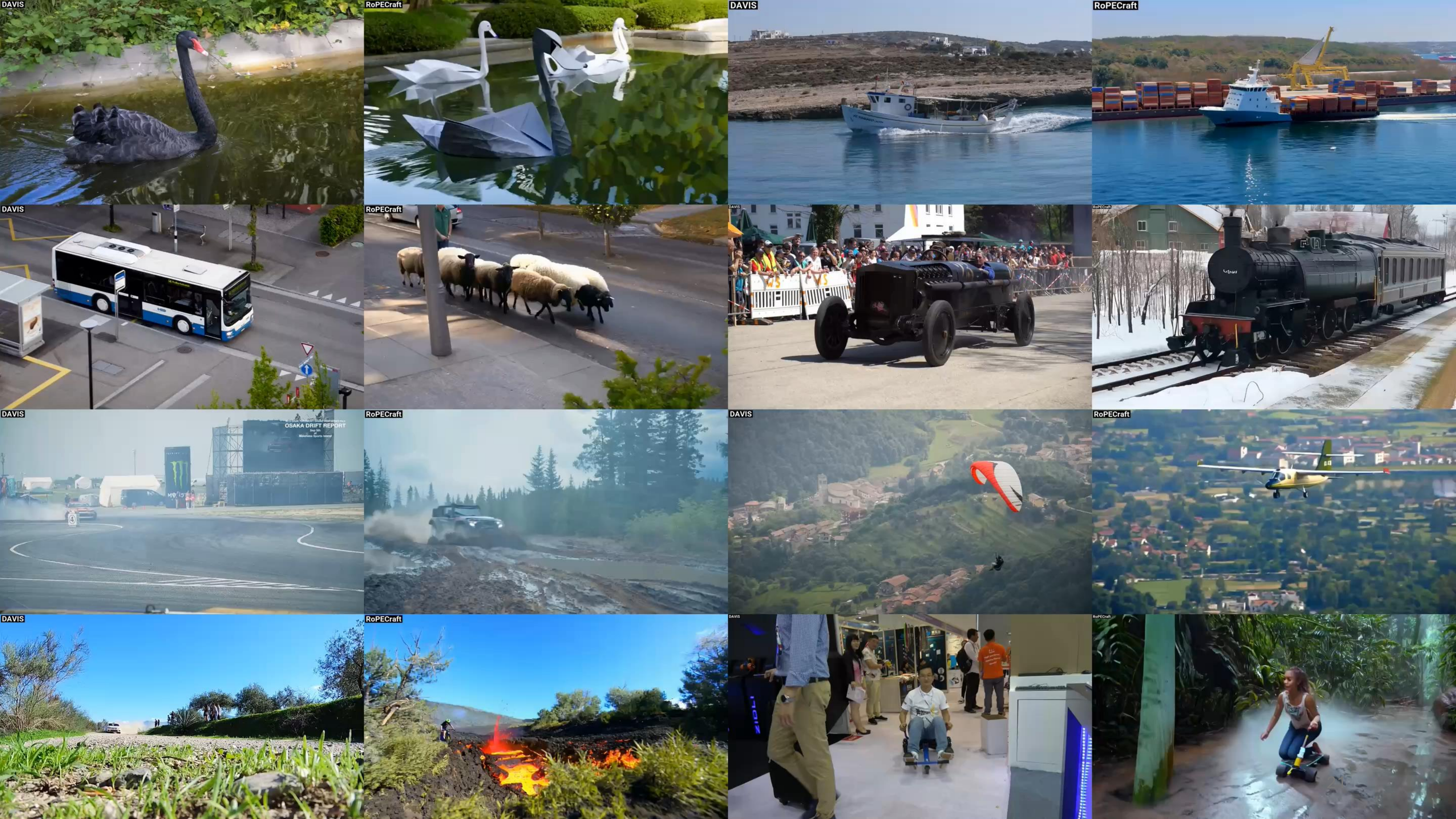
Qualitative results



Qualitative results



Qualitative results


DAVIS
RoPECraft
DAVIS
RoPECraft
DAVIS
RoPECraft
DAVIS
RoPECraft
OSAKA DRIFT REPORT
DAVIS
RoPECraft
DAVIS
RoPECraft

Thanks!