STRUCT Group Paper Reading

Taming Flow-based I2V Models for Creative Video Editing

arXiv 2025

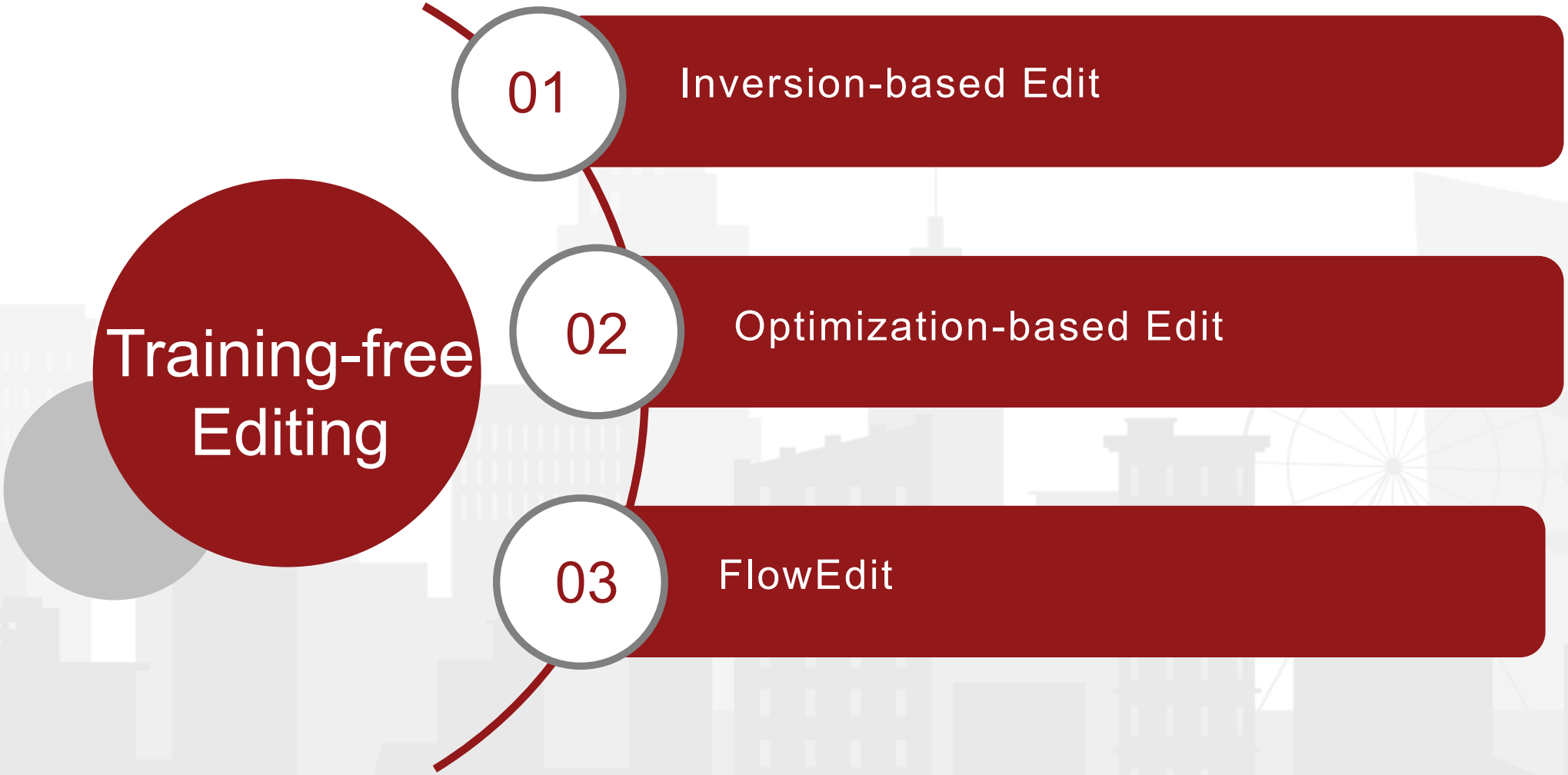
Xianghao Kong, Hansheng Chen, Yuwei Guo, Lvmin Zhang, Gordon Wetzstein, Maneesh Agrawala, Anyi Rao

Presented by Junyi Fan 2025.11.23

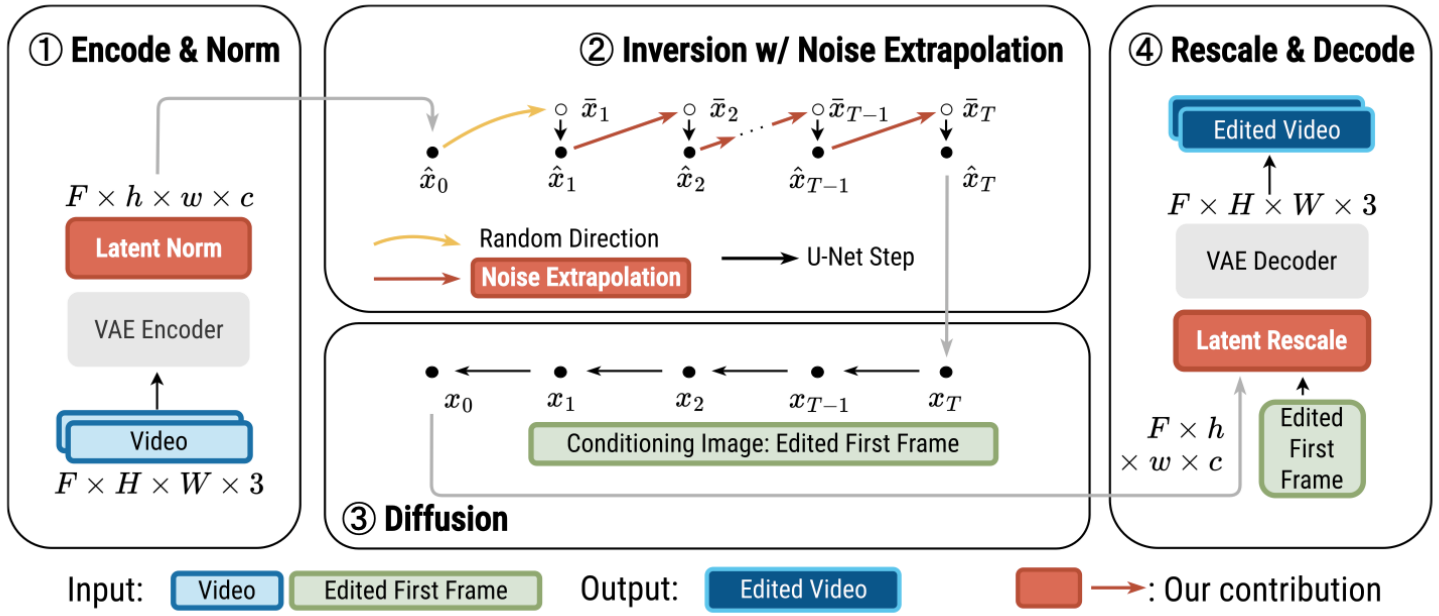
Video Editing: Task Formulation

- Task formulation:
  - Input:
    - Source video with L frames: $x^{src} = \{x_i^{src}\}_{i=\mathrm{j}sL}$
    - Edited first frame: x_{j}^{edit}
  - Output:
    - $x^{tar} = \{x_i^{tar}\}_{i=\mathrm{j}sL}$ such that propagate the modifications along the temporal dimension while maintaining overall structure and motion consistency with the source video

Background: Traning-free Visual Editing



Inversion-based Edit: VideoShop



a. Noise Extrapolation: to solve the inaccuracy of EDM inversion

b. Latent Normalization and Rescaling

Xiang Fan, Anand Bhattad, and Ranjay Krishna. "Videoshop: Localized Semantic Video Editing with Noise-Extrapolated Diffusion Inversion", in Proc. ECCV, 2024.

Inversion-based Edit: VideoShop

a. Noise Extrapolation: to solve the inaccuracy of EDM inversion

Inversion-based Edit: VideoShop

a. Noise Extrapolation: to solve the inaccuracy of EDM inversion
    - Inversion in EDM framework

Inversion-based Edit: VideoShop

a. Noise Extrapolation: to solve the inaccuracy of EDM inversion

- Inversion in EDM framework
  - Denoise step from $x_{t\mathtt{dj}} \rightarrow x_t$

$$x_t = x_{t+1} + \underbrace{\frac{\sigma_t - \sigma_{t+1}}{\sigma_{t+1}} \left(x_{t+1} - \overbrace{\left(c_{\text{skip}}^{t+1} x_{t+1} + c_{\text{out}}^{t+1} F_\theta \left(c_{\text{in}}^{t+1} x_{t+1}; c_{\text{noise}}^{t+1} \right) \right)}^{\text{predicted } x_0} \right)}_{\text{noise removed at step } t} \quad (3)$$

Inversion-based Edit: VideoShop

a. Noise Extrapolation: to solve the inaccuracy of EDM inversion

- Inversion in EDM framework
  - Denoise step from $x_{t\mathtt{dj}} \rightarrow x_t$

$$x_t = x_{t+1} + \underbrace{\frac{\sigma_t - \sigma_{t+1}}{\sigma_{t+1}} \left(x_{t+1} - \overbrace{\left(c_{\text{skip}}^{t+1} x_{t+1} + c_{\text{out}}^{t+1} F_\theta \left(c_{\text{in}}^{t+1} x_{t+1}; c_{\text{noise}}^{t+1} \right) \right)}^{\text{predicted } x_0} \right)}_{\text{noise removed at step } t} \tag{3}$$

  - Rewrite to invert

$$\hat{x}_{t+1} = \frac{\sigma_{t+1} \hat{x}_t + (\sigma_t - \sigma_{t+1})\, c_{\text{out}}^{t+1} F_\theta \left(c_{\text{in}}^{t+1} \hat{x}_{t+1}; c_{\text{noise}}^{t+1} \right)}{(\sigma_t - \sigma_{t+1}) \left(1 - c_{\text{skip}}^{t+1} \right) + \sigma_{t+1}} \tag{4}$$

Inversion-based Edit: VideoShop

a. Noise Extrapolation: to solve the inaccuracy of EDM inversion

- Inversion in EDM framework
  - Denoise step from $x_{t\mathtt{dj}} \rightarrow x_t$

$$x_t = x_{t+1} + \underbrace{\frac{\sigma_t - \sigma_{t+1}}{\sigma_{t+1}} \left(x_{t+1} - \overbrace{\left(c_{\text{skip}}^{t+1} x_{t+1} + c_{\text{out}}^{t+1} F_\theta \left(c_{\text{in}}^{t+1} x_{t+1}; c_{\text{noise}}^{t+1} \right) \right)}^{\text{predicted } x_0} \right)}_{\text{noise removed at step } t} \tag{3}$$

  - Rewrite to invert

$$\hat{x}_{t+1} = \frac{\sigma_{t+1} \hat{x}_t + (\sigma_t - \sigma_{t+1})\, c_{\text{out}}^{t+1} F_\theta \left(c_{\text{in}}^{t+1} \hat{x}_{t+1}; c_{\text{noise}}^{t+1} \right)}{(\sigma_t - \sigma_{t+1}) \left(1 - c_{\text{skip}}^{t+1} \right) + \sigma_{t+1}} \tag{4}$$

  - Approximation

$$F_\theta \left(c_{\text{in}}^{t+1} \hat{x}_{t+1}; c_{\text{noise}}^{t+1} \right) \approx F_\theta \left(c_{\text{in}}^{t} \hat{x}_{t}; c_{\text{noise}}^{t+1} \right) \tag{5}$$

Inversion-based Edit: VideoShop

a. Noise Extrapolation: to solve the inaccuracy of EDM inversion

- Inversion in EDM framework
  - Denoise step from $x_{t\mathtt{dj}} \rightarrow x_t$

$$x_t = x_{t+1} + \underbrace{\frac{\sigma_t - \sigma_{t+1}}{\sigma_{t+1}} \left(x_{t+1} - \overbrace{\left(c_{\text{skip}}^{t+1} x_{t+1} + c_{\text{out}}^{t+1} F_\theta \left(c_{\text{in}}^{t+1} x_{t+1}; c_{\text{noise}}^{t+1} \right) \right)}^{\text{predicted } x_0} \right)}_{\text{noise removed at step } t} \tag{3}$$

  - Rewrite to invert

$$\hat{x}_{t+1} = \frac{\sigma_{t+1} \hat{x}_t + (\sigma_t - \sigma_{t+1}) \, c_{\text{out}}^{t+1} F_\theta \left(c_{\text{in}}^{t+1} \hat{x}_{t+1}; c_{\text{noise}}^{t+1} \right)}{(\sigma_t - \sigma_{t+1}) \left(1 - c_{\text{skip}}^{t+1} \right) + \sigma_{t+1}} \tag{4}$$

  - Approximation

$$F_\theta \left(c_{\text{in}}^{t+1} \hat{x}_{t+1}; c_{\text{noise}}^{t+1} \right) \approx F_\theta \left(c_{\text{in}}^{t} \hat{x}_{t}; c_{\text{noise}}^{t+1} \right) \tag{5}$$

  - Introducing a compounding approximation error!

Inversion-based Edit: VideoShop

a. Noise Extrapolation: to solve the inaccuracy of EDM inversion
- Inversion in EDM framework
- Inverting with noise extrapolation

Inversion-based Edit: VideoShop

a. Noise Extrapolation: to solve the inaccuracy of EDM inversion

- Inversion in EDM framework
- Inverting with noise extrapolation
  - Observation: near-linearity of x_t trajectory

$x_{t_i} \rightarrow x_i$ and $x_{t_i} \rightarrow x_i$

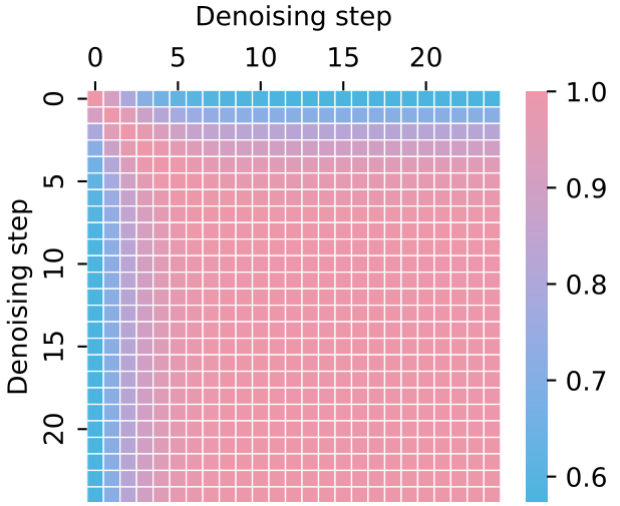


Figure. Cosine similarity matrix for pairs of latent vectors throughout the denoising process.

Inversion-based Edit: VideoShop

a. Noise Extrapolation: to solve the inaccuracy of EDM inversion
- Inversion in EDM framework
- Inverting with noise extrapolation
  - Observation: near-linearity of x_t trajectory
  - Noise extrapolation

$$\bar{x}_{t+1} \approx \begin{cases} \underbrace{\frac{\sigma_{t+1}}{\sigma_t}}_{\sim\mathcal{N}(0,\sigma_{t+1})} \overbrace{(\hat{x}_t - x_0)}^{\sim\mathcal{N}(0,\sigma_t)} + x_0 & (\sigma_t > \Sigma) \\ \mathcal{N}(0, \sigma_{t+1}) + x_0 & (\sigma_t \leq \Sigma) \end{cases} \tag{6}$$

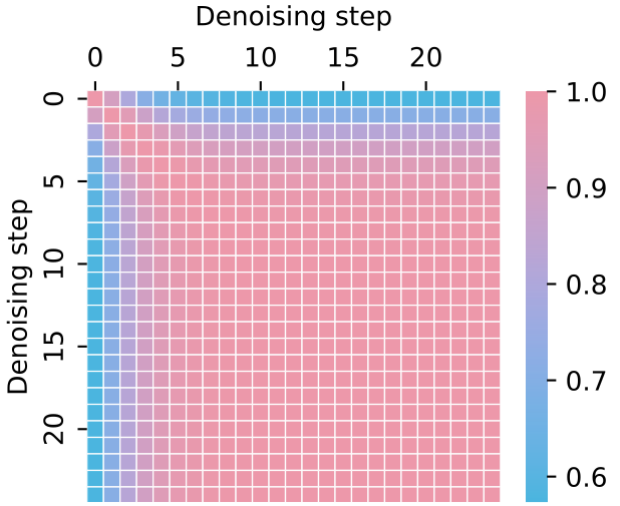


Figure. Cosine similarity matrix for pairs of latent vectors throughout the denoising process.

Inversion-based Edit: VideoShop

a. Noise Extrapolation: to solve the inaccuracy of EDM inversion
- Inversion in EDM framework
- Inverting with noise extrapolation
  - Observation: near-linearity of x_t trajectory
  - Noise extrapolation

$$\bar{x}_{t+1} \approx \begin{cases} \underbrace{\frac{\sigma_{t+1}}{\sigma_t}\overbrace{(\hat{x}_t - x_0)}^{\sim\mathcal{N}(0,\sigma_t)}}_{\sim\mathcal{N}(0,\sigma_{t+1})} + x_0 & (\sigma_t > \Sigma) \\ \mathcal{N}(0, \sigma_{t+1}) + x_0 & (\sigma_t \leq \Sigma) \end{cases} \tag{6}$$

$$\hat{x}_{t+1} = \frac{\sigma_{t+1}\hat{x}_t + (\sigma_t - \sigma_{t+1})\, c_{\text{out}}^{t+1} F_\theta\left(c_{\text{in}}^{t+1}\bar{x}_{t+1}; c_{\text{noise}}^{t+1}\right)}{(\sigma_t - \sigma_{t+1})\left(1 - c_{\text{skip}}^{t+1}\right) + \sigma_{t+1}} \tag{7}$$

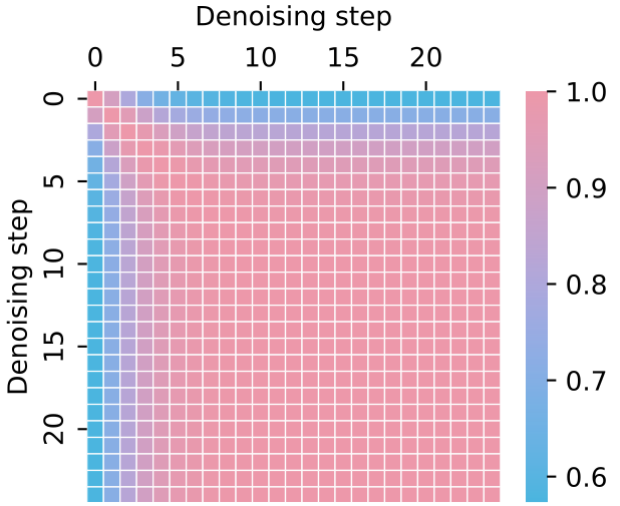


Figure. Cosine similarity matrix for pairs of latent vectors throughout the denoising process.

Inversion-based Edit: VideoShop

a. Noise Extrapolation: to solve the inaccuracy of EDM inversion

- Inversion in EDM framework
- Inverting with noise extrapolation
  - Observation: near-linearity of x_t trajectory
  - Noise extrapolation

$$\bar{x}_{t+1} \approx \begin{cases} \underbrace{\frac{\sigma_{t+1}}{\sigma_t}\overbrace{(\hat{x}_t - x_0)}^{\sim\mathcal{N}(0,\sigma_t)}}_{\sim\mathcal{N}(0,\sigma_{t+1})} + x_0 & (\sigma_t > \Sigma) \\ \mathcal{N}(0, \sigma_{t+1}) + x_0 & (\sigma_t \leq \Sigma) \end{cases} \tag{6}$$

$$\hat{x}_{t+1} = \frac{\sigma_{t+1}\hat{x}_t + (\sigma_t - \sigma_{t+1})\, c_{\text{out}}^{t+1} F_\theta\left(c_{\text{in}}^{t+1}\bar{x}_{t+1}; c_{\text{noise}}^{t+1}\right)}{(\sigma_t - \sigma_{t+1})\left(1 - c_{\text{skip}}^{t+1}\right) + \sigma_{t+1}} \tag{7}$$

  - Denoise x_T with target label

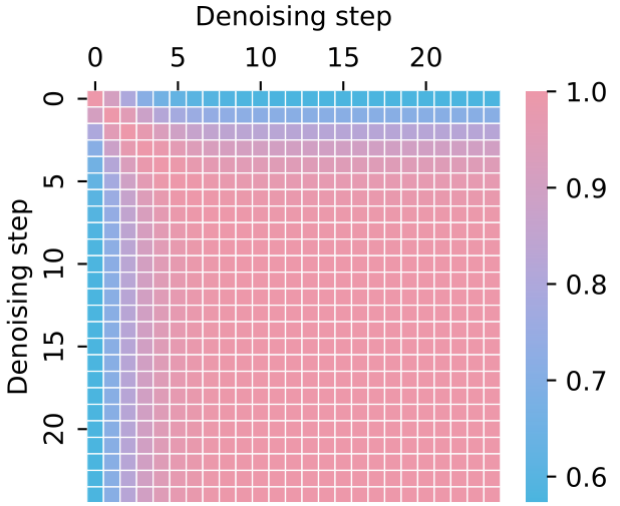
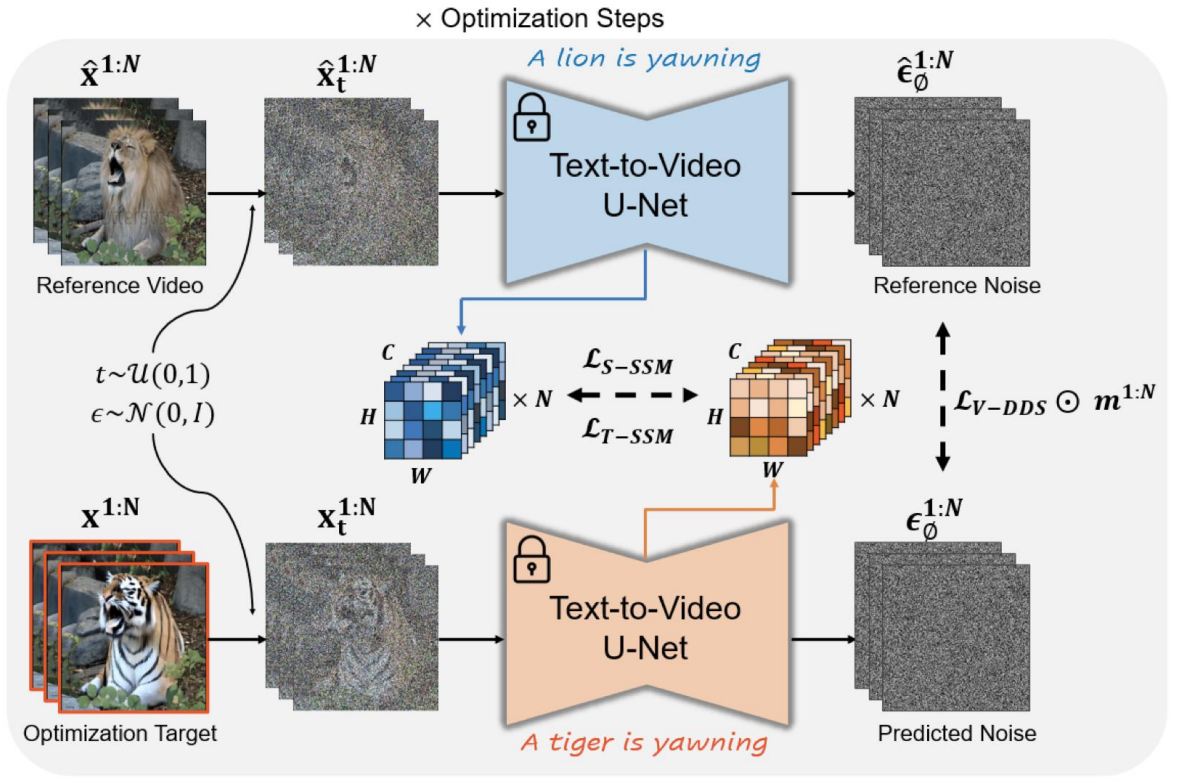


Figure. Cosine similarity matrix for pairs of latent vectors throughout the denoising process.

Optimization-based Edit: DreamMotion



a. Appearance Injection (V-DDS)

b. Structure Correction

c. Temporal Smoothing

Hyeonho Jeong, Jinho Chang, Geon Yeong Park, and Jong Chul Ye. "DreamMotion: Space-Time Self-Similar Score Distillation for Zero-Shot Video Editing", in Proc. ECCV, 2024.

Optimization-based Edit: DreamMotion

a. Appearance injection: applying score distillation to video

Optimization-based Edit: DreamMotion

a. Appearance injection: applying score distillation to video

- Image score distillation: align $x_i(\theta)$ with the target condition y by optimizing the diffusion training loss gradient

$$\mathcal{L}_{\mathrm{SDS}}(\theta; y) = \left\| \boldsymbol{\epsilon}_\phi^w(\boldsymbol{x}_t(\theta), t, y) - \boldsymbol{\epsilon} \right\|_2^2,$$

Optimization-based Edit: DreamMotion

a. Appearance injection: applying score distillation to video

- Image score distillation: align $x_i(\theta)$ with the target condition y by optimizing the diffusion training loss gradient

$$\mathcal{L}_{\mathrm{SDS}}(\theta; y) = \left\| \boldsymbol{\epsilon}_\phi^w(\boldsymbol{x}_t(\theta), t, y) - \boldsymbol{\epsilon} \right\|_2^2,$$

- SDS → DDS: incorporate a reference condition $\hat{y}$ and a reference image $\hat{x}_i$

$$\mathcal{L}_{\mathrm{DDS}}(\theta; y) = \left\| \boldsymbol{\epsilon}_\phi^w(\boldsymbol{x}_t(\theta), t, y) - \boldsymbol{\epsilon}_\phi^w(\hat{\boldsymbol{x}}_t, t, \hat{y}) \right\|_2^2.$$

Optimization-based Edit: DreamMotion

a. Appearance injection: applying score distillation to video

- Image score distillation: align $x_i(\theta)$ with the target condition y by optimizing the diffusion training loss gradient

$$\mathcal{L}_{\text{SDS}}(\theta; y) = \left\| \boldsymbol{\epsilon}_\phi^w(\boldsymbol{x}_t(\theta), t, y) - \boldsymbol{\epsilon} \right\|_2^2,$$

- SDS → DDS: incorporate a reference condition $\hat{y}$ and a reference image $\hat{x}_i$

$$\mathcal{L}_{\text{DDS}}(\theta; y) = \left\| \boldsymbol{\epsilon}_\phi^w(\boldsymbol{x}_t(\theta), t, y) - \boldsymbol{\epsilon}_\phi^w(\hat{\boldsymbol{x}}_t, t, \hat{y}) \right\|_2^2.$$

- Video score distillation with masked gradients
  - V-DDS:

$$\mathcal{L}_{\text{V-DDS}}(\theta; y) = \left\| \boldsymbol{\epsilon}_\phi^w(\boldsymbol{x}_t^{1:N}(\theta), t, y) - \boldsymbol{\epsilon}_\phi^w(\hat{\boldsymbol{x}}_t^{1:N}, t, \hat{y}) \right\|_2^2.$$

Optimization-based Edit: DreamMotion

a. Appearance injection: applying score distillation to video

- Image score distillation: align $x_i(\theta)$ with the target condition y by optimizing the diffusion training loss gradient

$$\mathcal{L}_{\text{SDS}}(\theta; y) = \left\| \epsilon_\phi^w(\boldsymbol{x}_t(\theta), t, y) - \boldsymbol{\epsilon} \right\|_2^2,$$

- SDS → DDS: incorporate a reference condition $\hat{y}$ and a reference image $\hat{x}_i$

$$\mathcal{L}_{\text{DDS}}(\theta; y) = \left\| \epsilon_\phi^w(\boldsymbol{x}_t(\theta), t, y) - \epsilon_\phi^w(\hat{\boldsymbol{x}}_t, t, \hat{y}) \right\|_2^2.$$

- Video score distillation with masked gradients
  - V-DDS:

$$\mathcal{L}_{\text{V-DDS}}(\theta; y) = \left\| \epsilon_\phi^w(\boldsymbol{x}_t^{1:N}(\theta), t, y) - \epsilon_\phi^w(\hat{\boldsymbol{x}}_t^{1:N}, t, \hat{y}) \right\|_2^2.$$

  - Masking gradients ($m^{\text{js}N}$ annotate the objects to be edited in each frame)

$$\nabla_0 \mathcal{L}_{\text{V-DDS}} \odot m^{1:N}.$$

Optimization-based Edit: DreamMotion

a. Appearance injection: applying score distillation to video

- Image score distillation: align $x_i(\theta)$ with the target condition y by optimizing the diffusion training loss gradient

$$\mathcal{L}_{\text{SDS}}(\theta; y) = \left\| \epsilon_\phi^w(\boldsymbol{x}_t(\theta), t, y) - \boldsymbol{\epsilon} \right\|_2^2,$$

- SDS → DDS: incorporate a reference condition $\hat{y}$ and a reference image $\hat{x}_i$

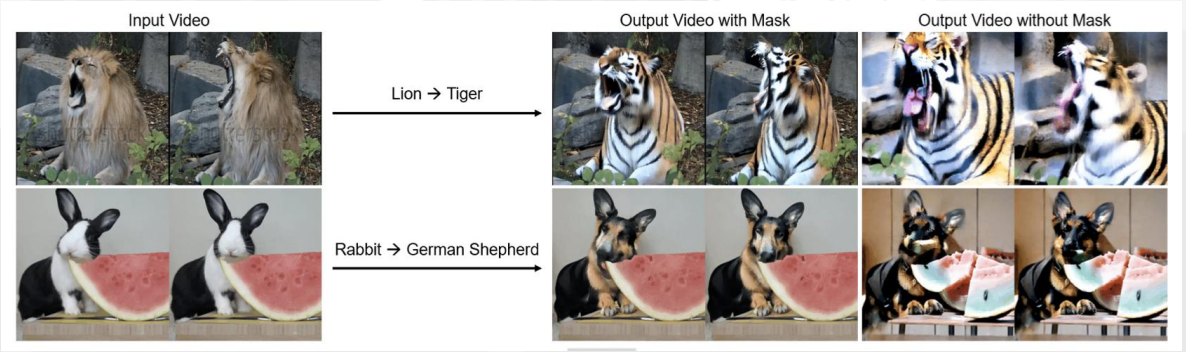
$$\mathcal{L}_{\text{DDS}}(\theta; y) = \left\| \epsilon_\phi^w(\boldsymbol{x}_t(\theta), t, y) - \epsilon_\phi^w(\hat{\boldsymbol{x}}_t, t, \hat{y}) \right\|_2^2.$$

- Video score distillation with masked gradients
  - V-DDS:

$$\mathcal{L}_{\text{V-DDS}}(\theta; y) = \left\| \epsilon_\phi^w(\boldsymbol{x}_t^{1:N}(\theta), t, y) - \epsilon_\phi^w(\hat{\boldsymbol{x}}_t^{1:N}, t, \hat{y}) \right\|_2^2.$$

  - Masking gradients

$$\nabla_\theta \mathcal{L}_{\text{V-DDS}} \odot m^{1:N}.$$



Optimization-based Edit: DreamMotion

b. Structure correction: spatial self-similarity matching

- Add identical noise
- Feed-forward to video diffusion's U-Net ϵ_θ for keys

Optimization-based Edit: DreamMotion

b. Structure correction: spatial self-similarity matching

- Add identical noise
- Feed-forward to video diffusion's U-Net ϵ_θ for keys
- Calculate spatial self-similarity map

$$SS^n_{i,j}(\boldsymbol{x}_t^{1:N}) = cos(K^n_i(x_t^{1:N}), K^n_j(x_t^{1:N})),$$

Optimization-based Edit: DreamMotion

b. Structure correction: spatial self-similarity matching

- Add identical noise
- Feed-forward to video diffusion's U-Net ϵ_θ for keys
- Calculate spatial self-similarity map

$$SS^n_{i,j}(\boldsymbol{x}_t^{1:N}) = cos(K^n_i(x_t^{1:N}), K^n_j(x_t^{1:N})),$$

$$\mathcal{L}_{\text{S-SSM}}(\boldsymbol{x}_t^{1:N}, \hat{\boldsymbol{x}}_t^{1:N}) = \frac{1}{N}\sum_{n=1}^{N}\left\| SS^n(\boldsymbol{x}_t^{1:N}) - SS^n(\hat{\boldsymbol{x}}_t^{1:N})\right\|_2^2,$$

Optimization-based Edit: DreamMotion

c. Temporal smoothing: temporal self-similarity matching

- Add identical noise
- Feed-forward to video diffusion's U-Net ϵ_θ for keys

Optimization-based Edit: DreamMotion

c. Temporal smoothing: temporal self-similarity matching

- Add identical noise
- Feed-forward to video diffusion's U-Net ϵ_θ for keys
- Spatial marginal mean

$$M[K(\boldsymbol{x}_t^{1:N})] = \frac{1}{H \cdot W} \sum_{i=1}^{H \cdot W} K_i(\boldsymbol{x}_t^{1:N}),$$

Optimization-based Edit: DreamMotion

c. Temporal smoothing: temporal self-similarity matching

- Add identical noise
- Feed-forward to video diffusion's U-Net ϵ_θ for keys
- Spatial marginal mean

$$M[K(\boldsymbol{x}_t^{1:N})] = \frac{1}{H \cdot W} \sum_{i=1}^{H \cdot W} K_i(\boldsymbol{x}_t^{1:N}),$$

- Calculate temporal self-similarity map

$$TS_{i,j}(\boldsymbol{x}_t^{1:N}) = cos(M_i[K(\boldsymbol{x}_t^{1:N})], M_j[K(\boldsymbol{x}_t^{1:N})]),$$

Optimization-based Edit: DreamMotion

c. Temporal smoothing: temporal self-similarity matching

- Add identical noise
- Feed-forward to video diffusion's U-Net ϵ_θ for keys
- Spatial marginal mean

$$M[K(\boldsymbol{x}_t^{1:N})] = \frac{1}{H \cdot W} \sum_{i=1}^{H \cdot W} K_i(\boldsymbol{x}_t^{1:N}),$$

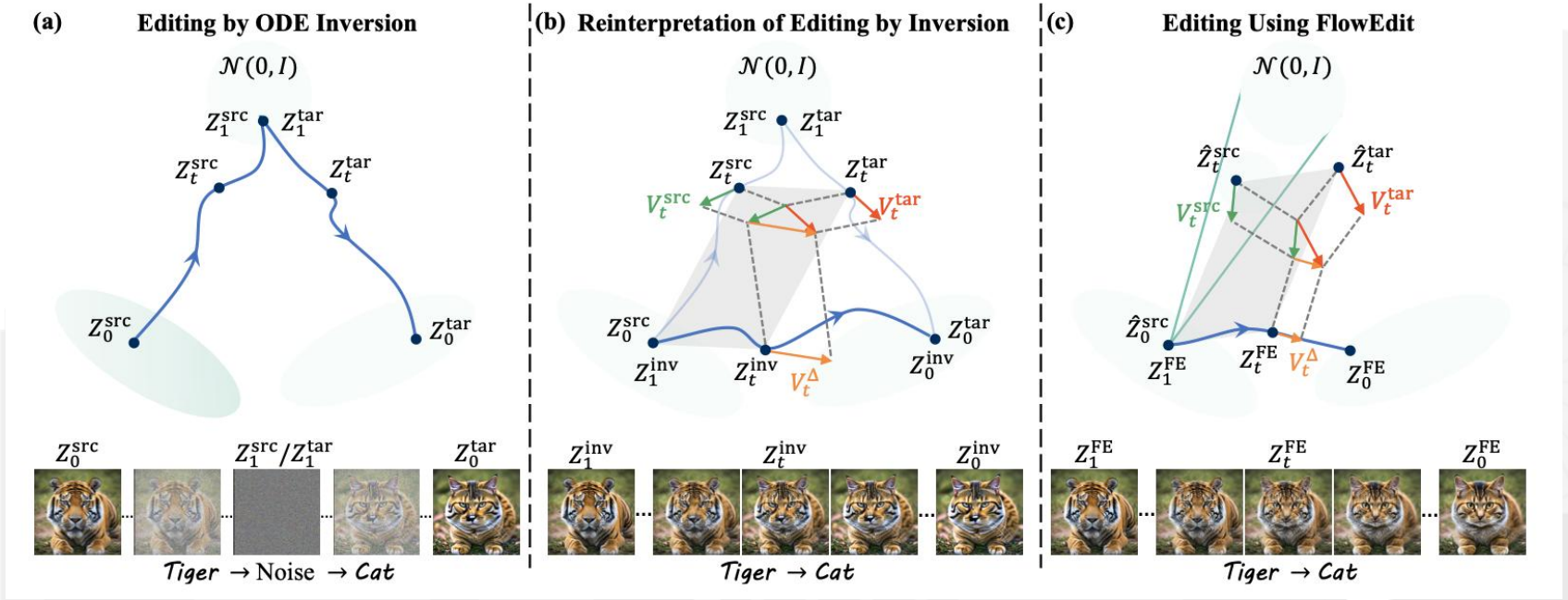
- Calculate temporal self-similarity map

$$TS_{i,j}(\boldsymbol{x}_t^{1:N}) = cos(M_i[K(\boldsymbol{x}_t^{1:N})], M_j[K(\boldsymbol{x}_t^{1:N})]),$$

$$\mathcal{L}_{\text{T-SSM}}(\boldsymbol{x}_t^{1:N}, \hat{\boldsymbol{x}}_t^{1:N}) = \left\| TS(\boldsymbol{x}_t^{1:N}) - TS(\hat{\boldsymbol{x}}_t^{1:N}) \right\|_2^2.$$

FlowEdit

- FlowEdit: Inversion-Free Text-Based Editing Using Pre-Trained Flow Models
  - Editing by inversion (a)
  - Reinterpretation of editing by inversion (b)

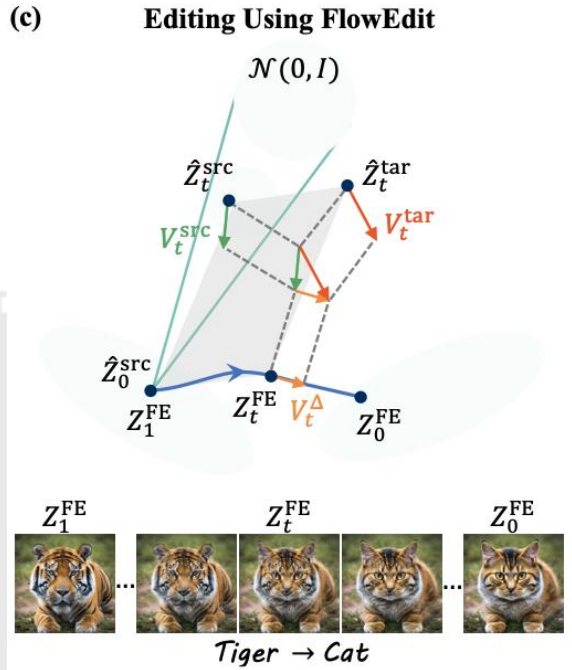


Vladimir Kulikov, Matan Kleiner, Inbar Huberman-Spiegelglas, and Tomer Michaeli. "FlowEdit: Inversion-Free Text-Based Editing Using Pre-Trained Flow Models", arXiv:2412.08629 [cs.CV], 2024.

FlowEdit

- FlowEdit: Inversion-Free Text-Based Editing Using Pre-Trained Flow Models
  - Editing by inversion (a)
  - Reinterpretation of editing by inversion (b)
  - Editing using flow-edit (c)
    - Shorter direct path

$$\hat{Z}_t^{\text{src}} = (1-t)Z_0^{\text{src}} + tN_t,$$

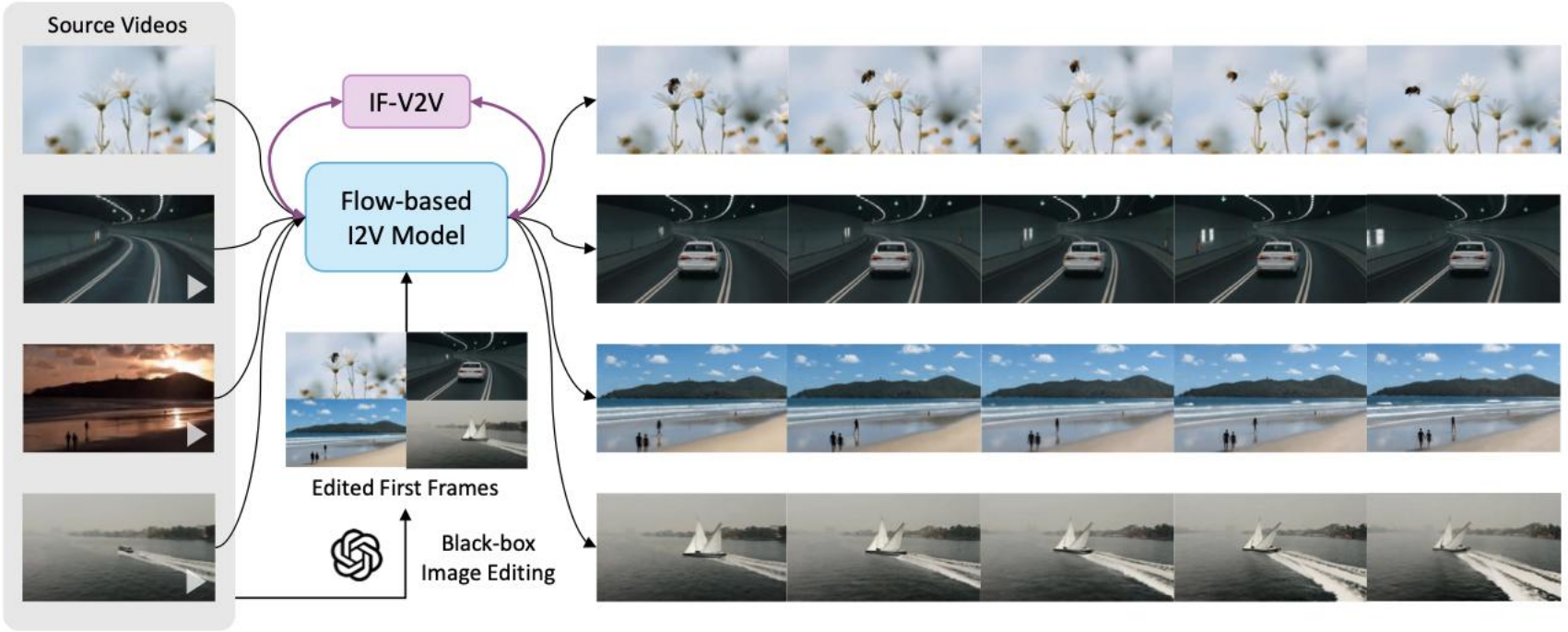


Algorithm 1 Simplified algorithm for FlowEdit

Input: real image X^{src}, $\{t_i\}_{i=0}^{T}$, $n_{\max}$, n_{avg}
Output: edited image X^{tar}
Init: $Z_{t_{\max}}^{\text{FE}} = X_0^{\text{src}}$
for $i = n_{\max}$ **to** 1 **do**
  $N_{t_i} \sim \mathcal{N}(0, 1)$
  $Z_{t_i}^{\text{src}} \leftarrow (1-t_i)X^{\text{src}} + t_i N_{t_i}$
  $Z_{t_i}^{\text{tar}} \leftarrow Z_{t_i}^{\text{FE}} + Z_{t_i}^{\text{src}} - X^{\text{src}}$
  $V_{t_i}^{\Delta} \leftarrow V^{\text{tar}}(Z_{t_i}^{\text{tar}}, t_i) - V^{\text{src}}(Z_{t_i}^{\text{src}}, t_i)$
  $Z_{t_{i-1}}^{\text{FE}} \leftarrow Z_{t_i}^{\text{FE}} + (t_{i-1} - t_i)V_{t_i}^{\Delta}$
end for
Return: $Z_0^{\text{FE}} = X_0^{\text{tar}}$

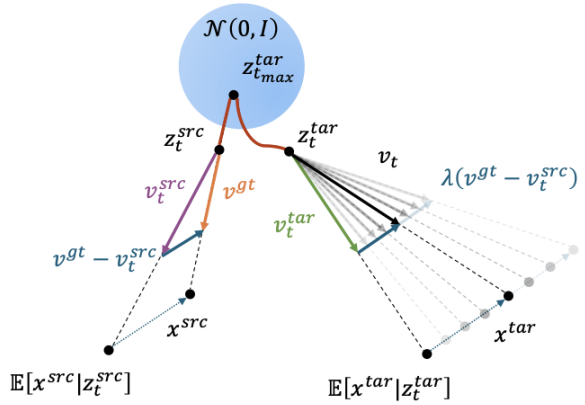
Optionally average n_{avg} samples

Methodology: Overview

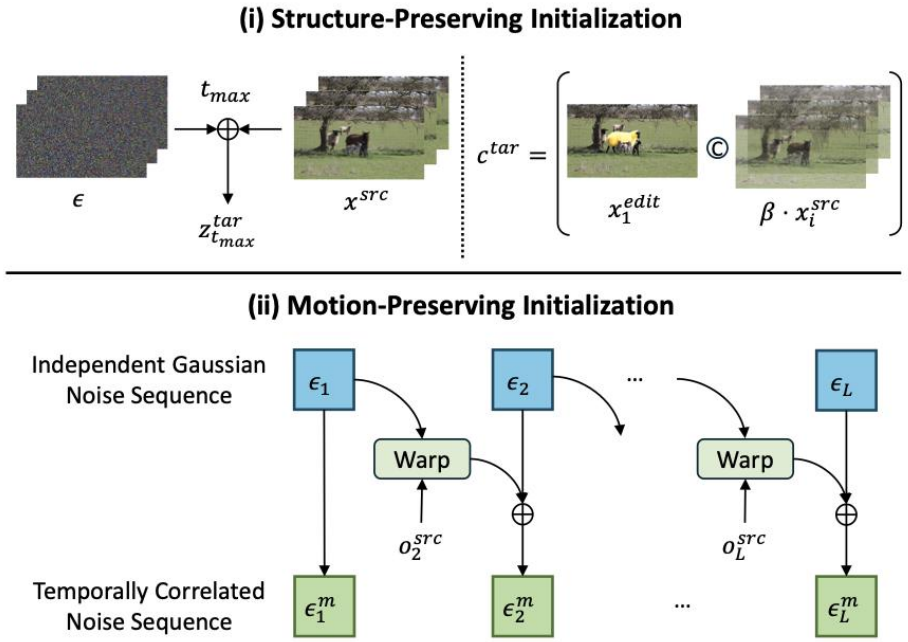


Contribution: Inversion-Free Video to Video editing (IF-V2V)

Methodology: Framework



a. Vector Field Rectification With Sample Deviation (VFR-SD)



b. Sturcture-And-Motion-Preserving Initialization (SMPI)

$$d(t_a,t_b)=\sum_{t=t_a}^{t_b-\Delta t}\|v_t^{tar}-v_{t+\Delta t}^{tar}\|_1,$$

c. Deviation Caching (D-CACHE)

Methodology: VFR-SD

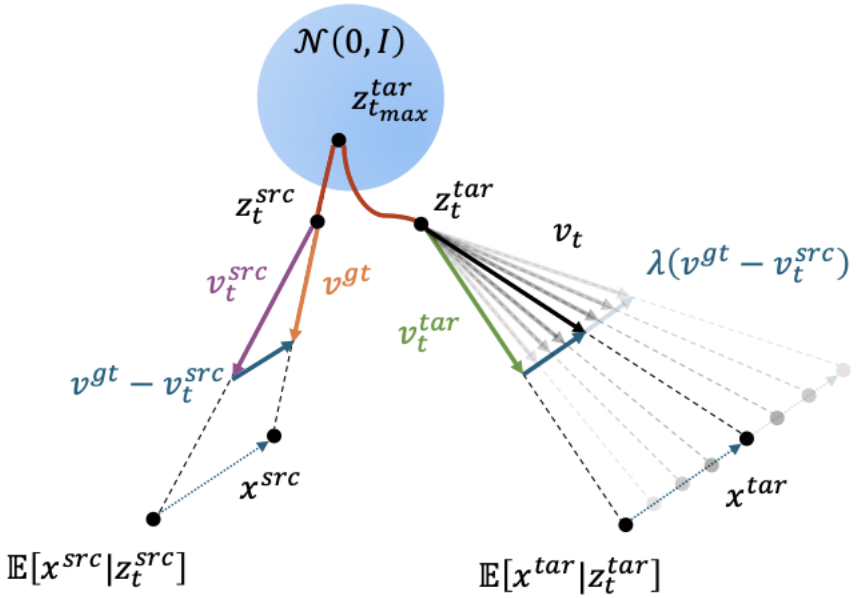
a. Vector field rectification with sample deviation (VFR-SD)

Methodology: VFR-SD

a. Vector field rectification with sample deviation (VFR-SD)

- Target ODE:

$$\mathrm{d}z_t^{tar} = v(z_t^{tar}, t, c^{tar})\mathrm{d}t,$$



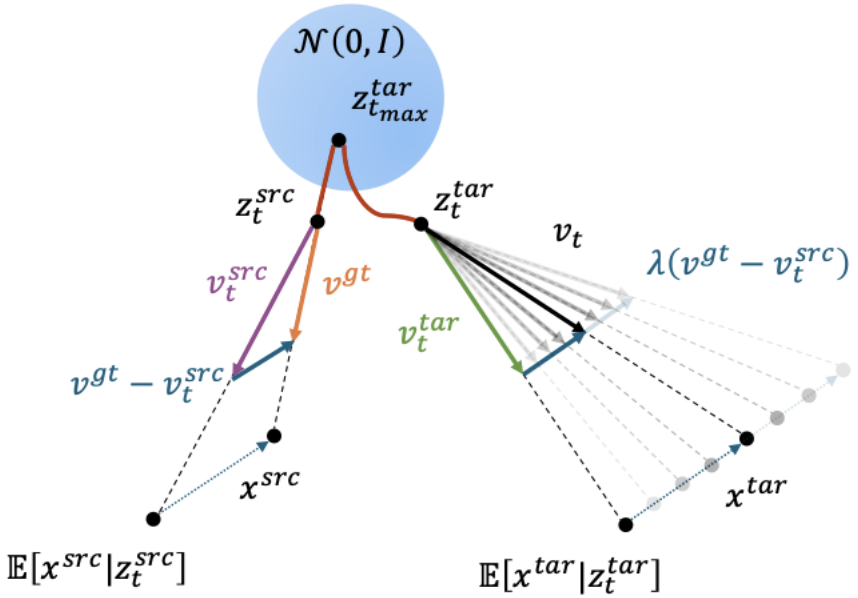
Methodology: VFR-SD

a. Vector field rectification with sample deviation (VFR-SD)

- Target ODE:

$$\mathrm{d}z_t^{tar} = v(z_t^{tar}, t, c^{tar})\mathrm{d}t,$$

- Solution v_t^{tar} Less information from x^{src}!



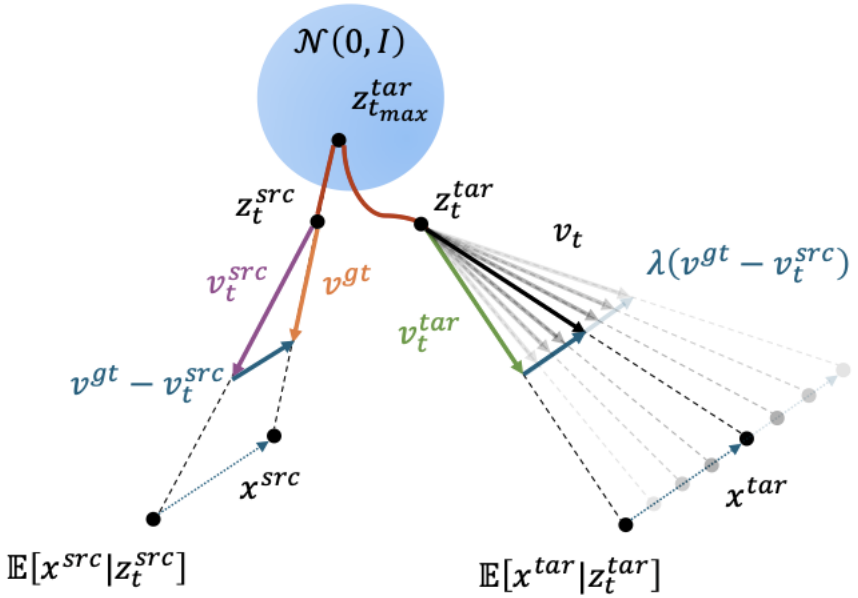
Methodology: VFR-SD

a. Vector field rectification with sample deviation (VFR-SD)

- Target ODE:

$$\mathrm{d}z_t^{tar} = v(z_t^{tar}, t, c^{tar})\mathrm{d}t,$$

- Solution v_t^{tar} Less information from x^{src}!
- Observation: v^{gt} contains both information from x^{src} and c^{src}



Methodology: VFR-SD

a. Vector field rectification with sample deviation (VFR-SD)

- Target ODE:

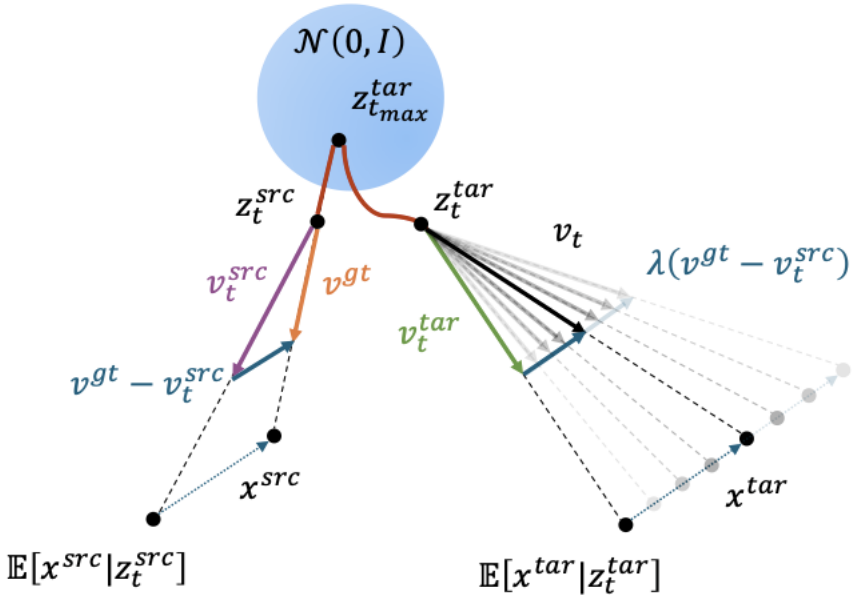
$$\mathrm{d}z_t^{tar} = v(z_t^{tar}, t, c^{tar})\mathrm{d}t,$$

- Solution v_t^{tar} Less information from x^{src}!
- Observation: v^{gt} contains both information from x^{src} and c^{src}

- Parallel ODE:

$$\mathrm{d}z_t^{src} = v(z_t^{src}, t, c^{src})\mathrm{d}t.$$

- Solution v_t^{src} only contains information from c^{src}



Methodology: VFR-SD

a. Vector field rectification with sample deviation (VFR-SD)

- Target ODE:

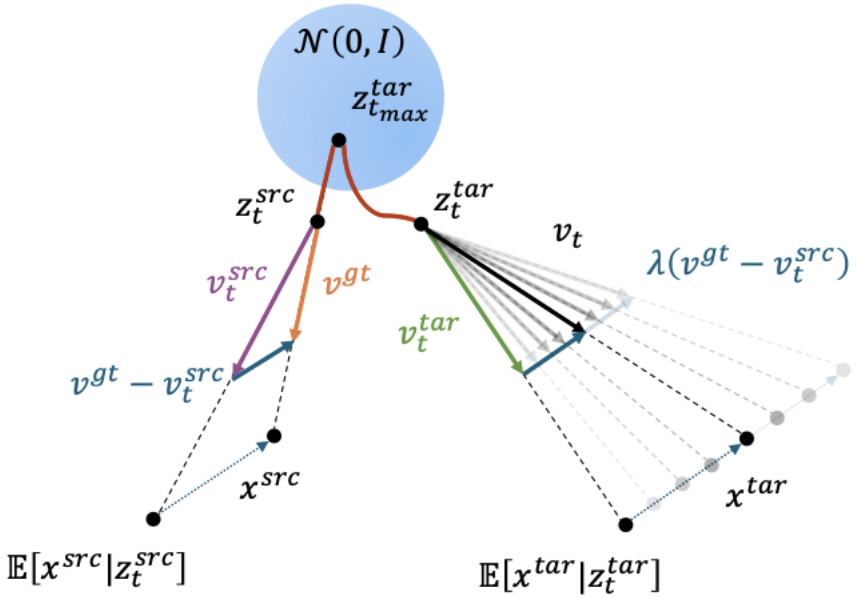
$$\mathrm{d}z_t^{tar} = v(z_t^{tar}, t, c^{tar})\mathrm{d}t,$$

  - Solution v_t^{tar} Less information from x^{src}!
  - Observation: v^{gt} contains both information from x^{src} and c^{src}

- Parallel ODE:

$$\mathrm{d}z_t^{src} = v(z_t^{src}, t, c^{src})\mathrm{d}t.$$

  - Solution v_t^{src} only contains information from c^{src}
  - Rectification vector: $v^{gt} - v_t^{src}$



Methodology: VFR-SD

a. Vector field rectification with sample deviation (VFR-SD)

- Target ODE:

$$\mathrm{d}z_t^{tar} = v(z_t^{tar}, t, c^{tar})\mathrm{d}t,$$

  - Solution v_t^{tar} Less information from x^{src}!
  - Observation: v^{gt} contains both information from x^{src} and c^{src}

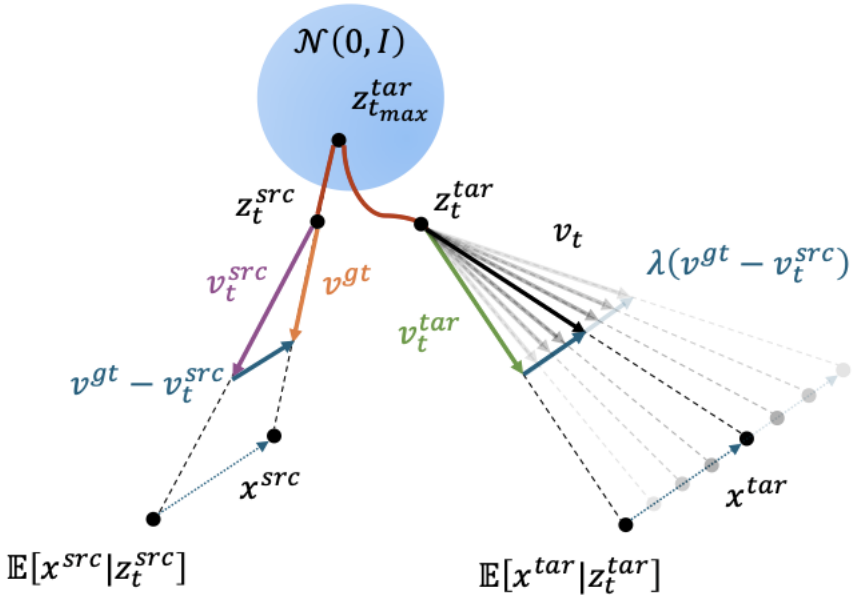
- Parallel ODE:

$$\mathrm{d}z_t^{src} = v(z_t^{src}, t, c^{src})\mathrm{d}t.$$

  - Solution v_t^{src} only contains information from c^{src}
  - Rectification vector: $v^{gt} - v_t^{src}$

- Apply to v_t^{tar} (param λ serves as the rectification scale, set to 1.0 in experiments)

$$v_t = v_t^{tar} + \lambda(v^{gt} - v_t^{src}),$$



Methodology: VFR-SD

a. Vector field rectification with sample deviation (VFR-SD)

ALGORITHM 1: Vector Field Rectification with Sample Deviation (VFR- SD, §3.3)

Input: Source video x^{src}, source condition c^{src}, target condition c^{tar}, flow model v_θ, initial timestep t_{max}, rectification scale λ.

Output: Edited video x^{tar}.

$\epsilon \sim \mathcal{N}(0, I)$

$z^{tar}_{t_{max}}, z^{src}_{t_{max}} \leftarrow (1 - t_{max})x^{src} + t_{max}\epsilon$ `// Latents initialization.`

$v^{gt} \leftarrow \epsilon - x^{src}$

`// Numerically solve the parallel ODEs.`

for $t \leftarrow t_{max}$ *downto* 0 **do**

  $v^{tar}_t \leftarrow v_\theta(z^{tar}_t, t, \boldsymbol{c}^{tar})$ `// Predict the target denoising vector.`

  $v^{src}_t \leftarrow v_\theta(z^{src}_t, t, \boldsymbol{c}^{src})$ `// Predict the source denoising vector.`

  $v_t \leftarrow v^{tar}_t + \lambda(v^{gt} - v^{src}_t)$ `// Rectification.`

  $z^{tar}_{t-\Delta t} \leftarrow \texttt{solver}_{t \to t-\Delta t}(z^{tar}_t, v_t)$ `// Update target latents accordingly.`

  $z^{src}_{t-\Delta t} \leftarrow \texttt{solver}_{t \to t-\Delta t}(z^{src}_t, v^{gt}_t)$ `// Update source latents with GT vector.`

end

return $x^{tar} \leftarrow z^{tar}_0$

Methodology: SMPI

b. Sturcture-and-motion-preserving initialization (SMPI)
- Structure-preserving initialization

Methodology: SMPI

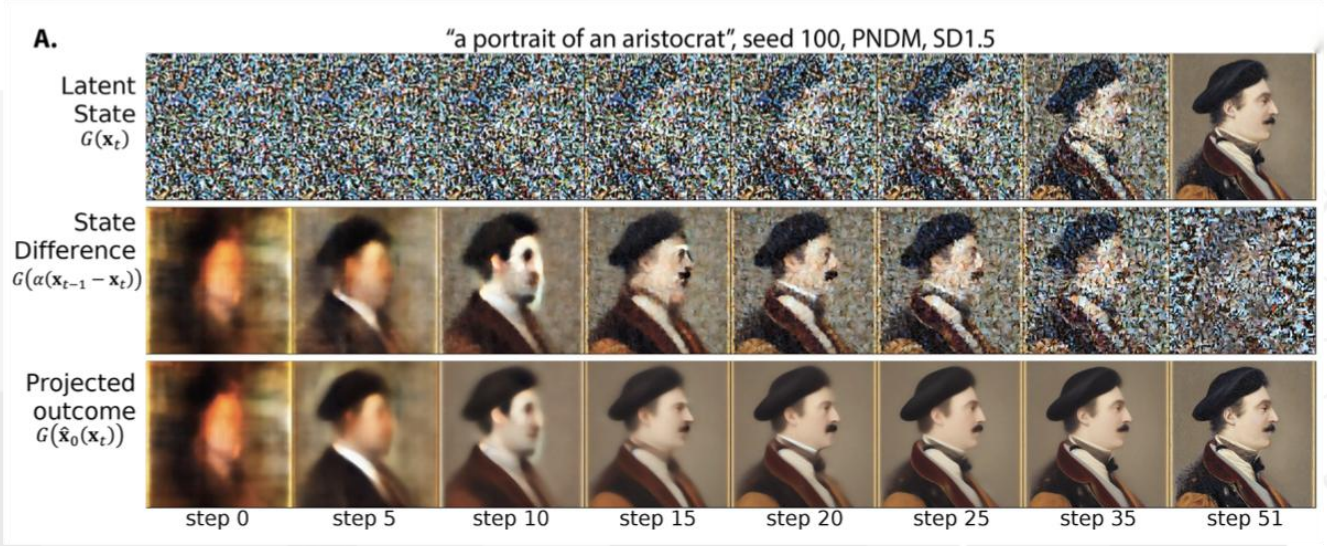
b. Sturcture-and-motion-preserving initialization (SMPI)

- Structure-preserving initialization
  - Observation-1: visual outlines are generated in the early stages of diffusion sampling and details at later timesteps

Methodology: SMPI

b. Sturcture-and-motion-preserving initialization (SMPI)

- Structure-preserving initialization
  - Observation-1: visual outlines are generated in the early stages of diffusion sampling and details at later timesteps
    - Figure. Characteristics of image generation by diffusion models

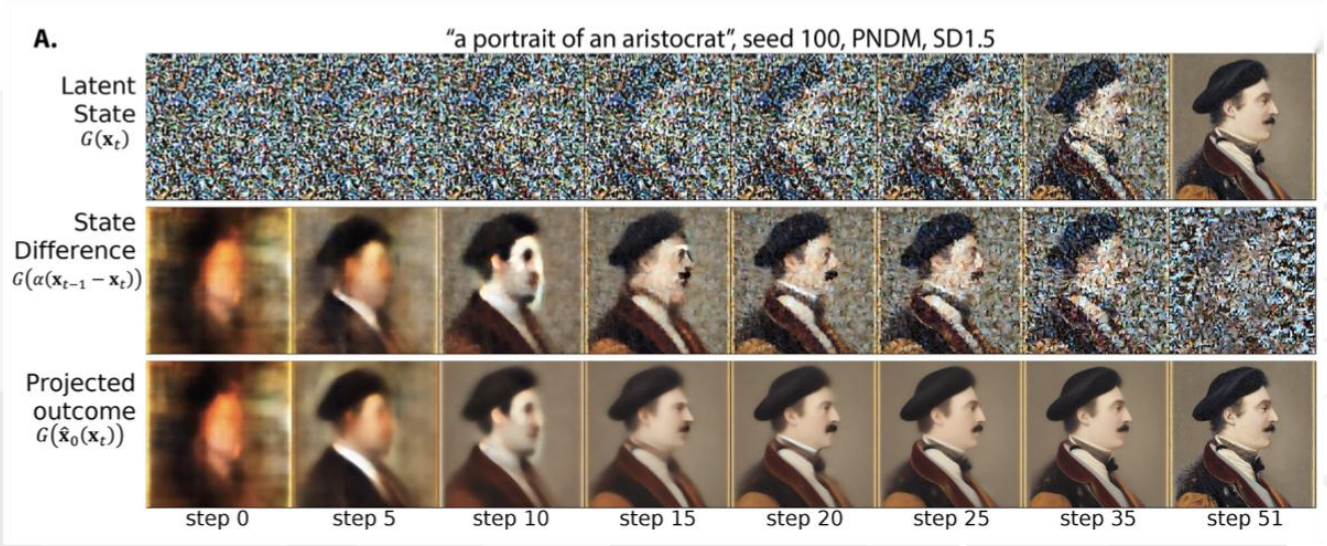


Binxu Wang, and John J. Vastola. "Diffusion Models Generate Images Like Painters: an Analytical Theory of Outline First, Details Later", arXiv:2303.02490 [cs.CV], 2023.

Methodology: SMPI

b. Sturcture-and-motion-preserving initialization (SMPI)

- Structure-preserving initialization
  - Observation-1: visual outlines are generated in the early stages of diffusion sampling and details at later timesteps
    - Figure. Characteristics of image generation by diffusion models



- Initial t_{max} < 1 (pure noise)

Methodology: SMPI

b. Sturcture-and-motion-preserving initialization (SMPI)

- Structure-preserving initialization
  - Observation-1: visual outlines are generated in the early stages of diffusion sampling and details at later timesteps
    - Initial t_{max} < 1 (pure noise) (set to 0.95 in experiments)

$$z_{t_{max}}^{tar} = (1 - t_{max})x^{src} + t_{max}\epsilon,$$

Methodology: SMPI

b. Sturcture-and-motion-preserving initialization (SMPI)

- Structure-preserving initialization
  - Observation-1: visual outlines are generated in the early stages of diffusion sampling and details at later timesteps
    - Initial t_{max} < 1 (pure noise) (set to 0.95 in experiments)

$$z_{t_{max}}^{tar} = (1 - t_{max})x^{src} + t_{max}\epsilon,$$

  - Observation-2: condition c^{tar} in mainstream I2V models consists of the concatenation of the first frame and zero paddings to align with video length L

Methodology: SMPI

b. Sturcture-and-motion-preserving initialization (SMPI)

- Structure-preserving initialization
  - Observation-1: visual outlines are generated in the early stages of diffusion sampling and details at later timesteps
    - Initial t_{max} < 1 (pure noise) (set to 0.95 in experiments)

$$z_{t_{max}}^{tar} = (1 - t_{max})x^{src} + t_{max}\epsilon,$$

  - Observation-2: condition c^{tar} in mainstream I2V models consists of the concatenation of the first frame and zero paddings to align with video length L
    - Compose following c^{tar} to encode information, β is a small embedding scale (set to 0.025 in experiments)

$$c^{tar} = \texttt{concat}(x_1^{edit}, \beta\{x_i^{src}\}_{i=2}^{L}),$$

Methodology: SMPI

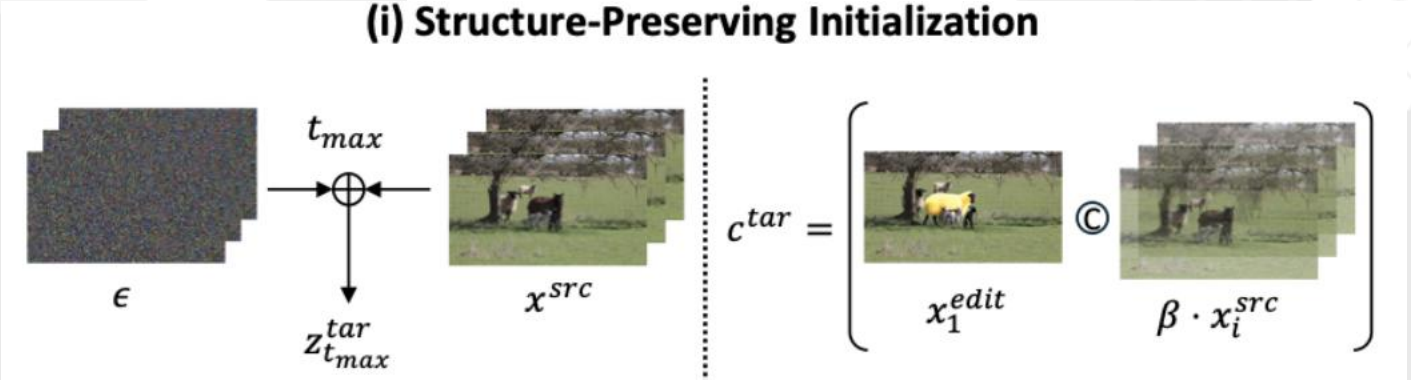
b. Sturcture-and-motion-preserving initialization (SMPI)

- Structure-preserving initialization
  - Observation-1: visual outlines are generated in the early stages of diffusion sampling and details at later timesteps
    - Initial t_{max} < 1 (pure noise) (set to 0.95 in experiments)

$$z_{t_{max}}^{tar} = (1 - t_{max})x^{src} + t_{max}\epsilon,$$

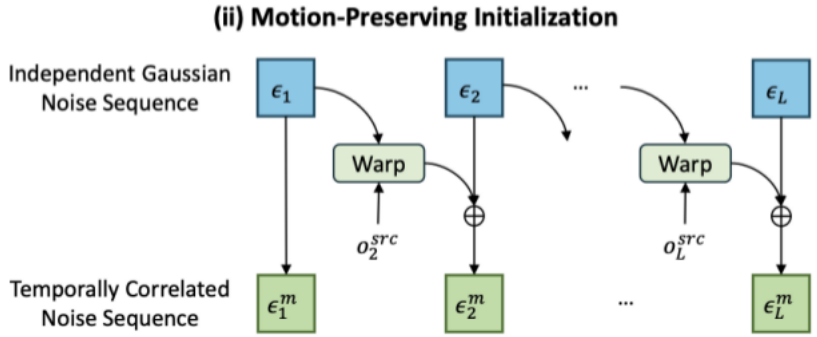
  - Observation-2: condition c^{tar} in mainstream I2V models consists of the concatenation of the first frame and zero paddings to align with video length L
    - Compose following c^{tar} to encode information, β is a small embedding scale (set to 0.025 in experiments)

$$c^{tar} = \texttt{concat}(x_1^{edit}, \beta\{x_i^{src}\}_{i=2}^{L}),$$



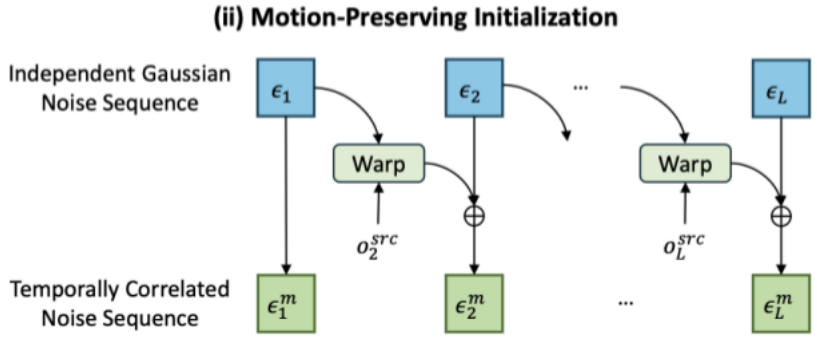
Methodology: SMPI

b. Sturcture-and-motion-preserving initialization (SMPI)
- Structure-preserving initialization
- Motion-preserving initialization



Methodology: SMPI

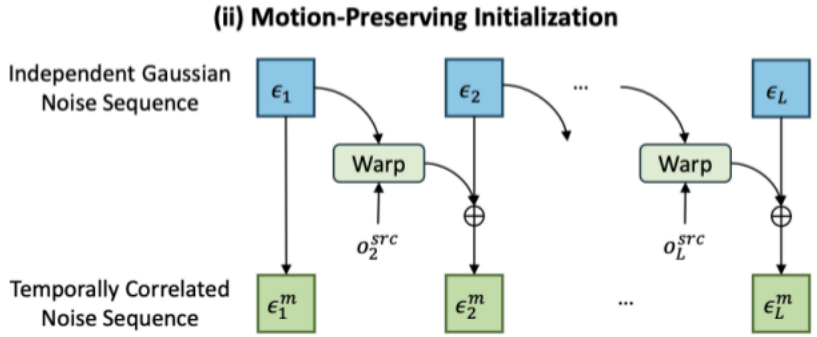
b. Sturcture-and-motion-preserving initialization (SMPI)
- Structure-preserving initialization
- Motion-preserving initialization
  - Noise-warping method



Ryan Burgert, Yuancheng Xu, Wenqi Xian, et al. "Go-with-the-Flow: Motion-Controllable Video Diffusion Models Using Real-Time Warped Noise", in Proc. CVPR, 2025.

Methodology: SMPI

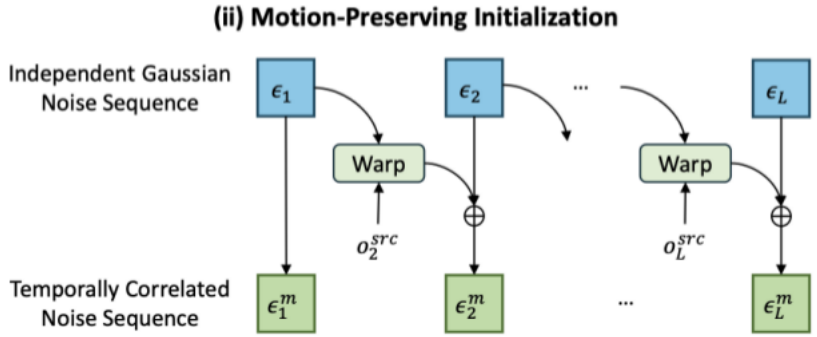
b. Sturcture-and-motion-preserving initialization (SMPI)
- Structure-preserving initialization
- Motion-preserving initialization
  - Noise-warping method
    - Extract source video's optical flow $\{o_i^{src}\}_{i=\mathrm{k}}^{L}$
    - Sample independent Gaussian noise $\epsilon = \{\epsilon_i\}_{i=\mathrm{j}}^{L}$



Ryan Burgert, Yuancheng Xu, Wenqi Xian, et al. "Go-with-the-Flow: Motion-Controllable Video Diffusion Models Using Real-Time Warped Noise", in Proc. CVPR, 2025.

Methodology: SMPI

b. Sturcture-and-motion-preserving initialization (SMPI)
- Structure-preserving initialization
- Motion-preserving initialization
  - Noise-warping method
    - Extract source video's optical flow $\{o_i^{src}\}_{i=\mathrm{k}}^{L}$
    - Sample independent Gaussian noise $\epsilon = \{\epsilon_i\}_{i=\mathrm{j}}^{L}$
    - Modulate noise as follows:



$$\epsilon_1^m = \epsilon_1,$$

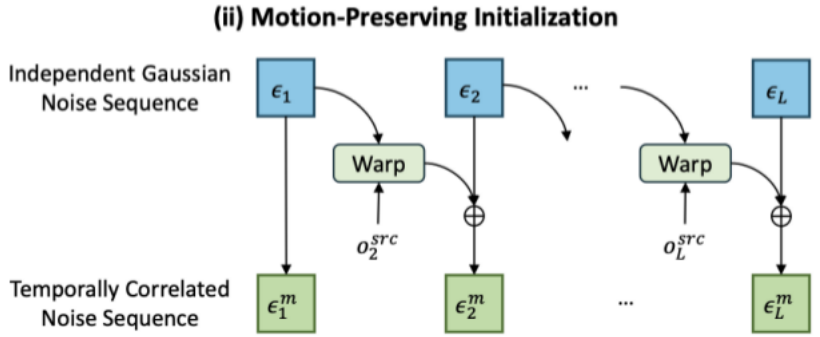
$$\epsilon_i^m = \frac{1}{\sqrt{(1-\alpha)^2+\alpha^2}}((1-\alpha)\cdot \mathtt{warp}(\epsilon_{i-1}, o_i^{src}) + \alpha\epsilon_i),$$

    - Warp: 2D warping operation according to the optical flow
    - Blending factor α (set to 0.95 in expriments): hyperparameter for temporal correlation degree

Ryan Burgert, Yuancheng Xu, Wenqi Xian, et al. "Go-with-the-Flow: Motion-Controllable Video Diffusion Models Using Real-Time Warped Noise", in Proc. CVPR, 2025.

Methodology: SMPI

b. Sturcture-and-motion-preserving initialization (SMPI)
- Structure-preserving initialization
- Motion-preserving initialization
  - Noise-warping method

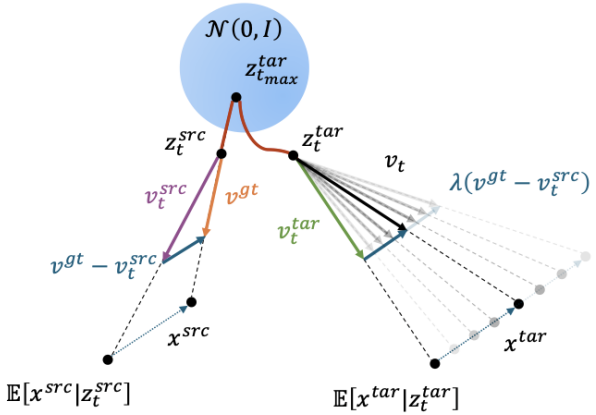


Prompt: A snowboarder racing down a snowy slope.

Prompt: A hot air ballon souring over a scenic village.

Methodology: D-CACHE

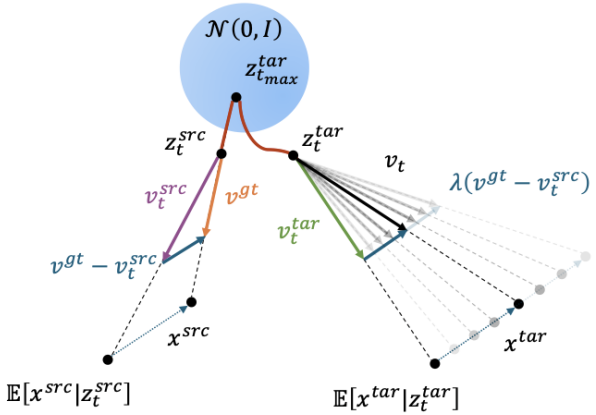
c. Deviation caching (D-CACHE)



Methodology: D-CACHE

c. Deviation caching (D-CACHE)

- Try to reduce the cost of calculating $v^{gt} - v_t^{src}$
  - v^{gt} is constant



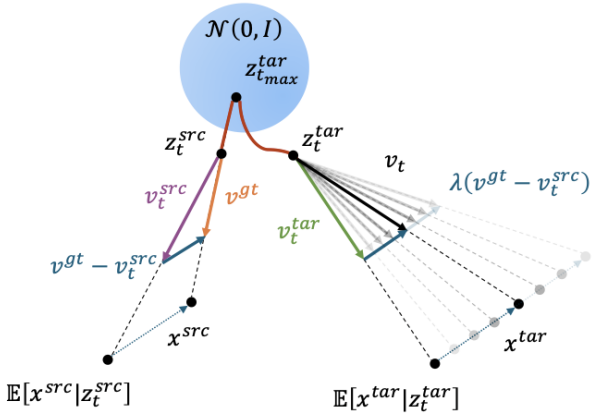
Methodology: D-CACHE

c. Deviation caching (D-CACHE)

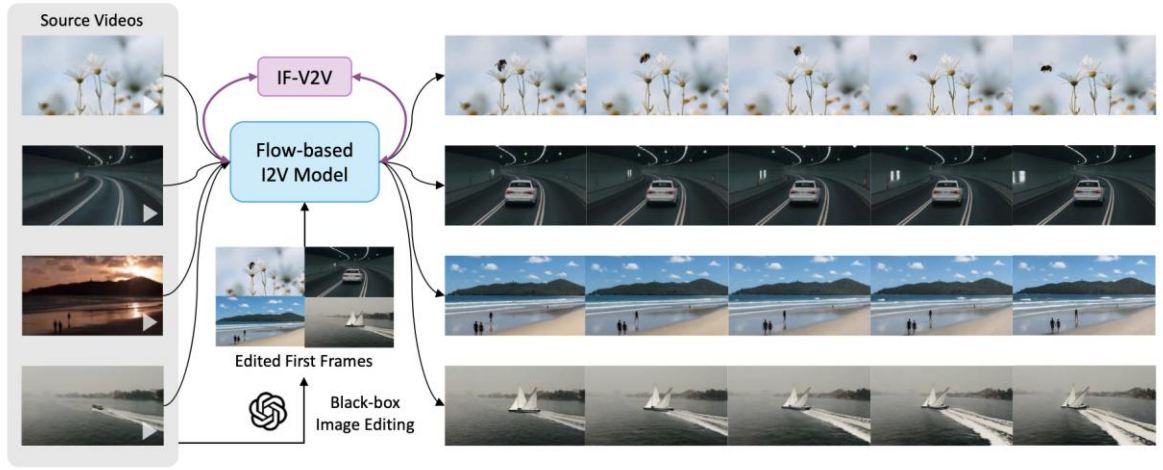
- Try to reduce the cost of calculating $v^{gt} - v_t^{src}$
  - v^{gt} is constant
  - Reuse $v^{gt} - v_t^{src}$ when the variation of v_t^{tar} is small
  - Def. cumulative variation:

$$d(t_a, t_b) = \sum_{t=t_a}^{t_b-\Delta t} \|v_t^{tar} - v_{t+\Delta t}^{tar}\|_1,$$

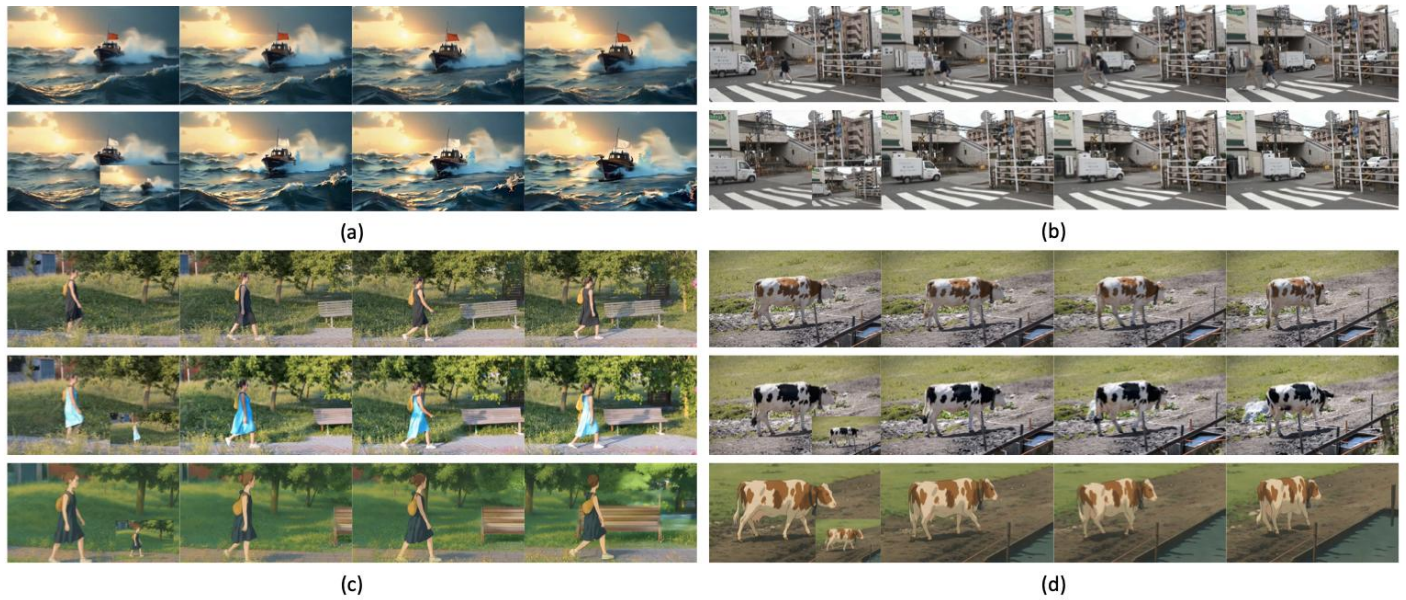
  - if $d(t_a, t_b) \leq$ threshold δ (set to 0.5 in experiments), reuse cached source denoising vector
  - else, predict and cache v_t^{src}



Experiments: Qualitative Results



(teaser) Object addition



(a) & (c.1) & (d.1) Attribute modification

(b) Object removal

(c.2) & (d.2) Stylization

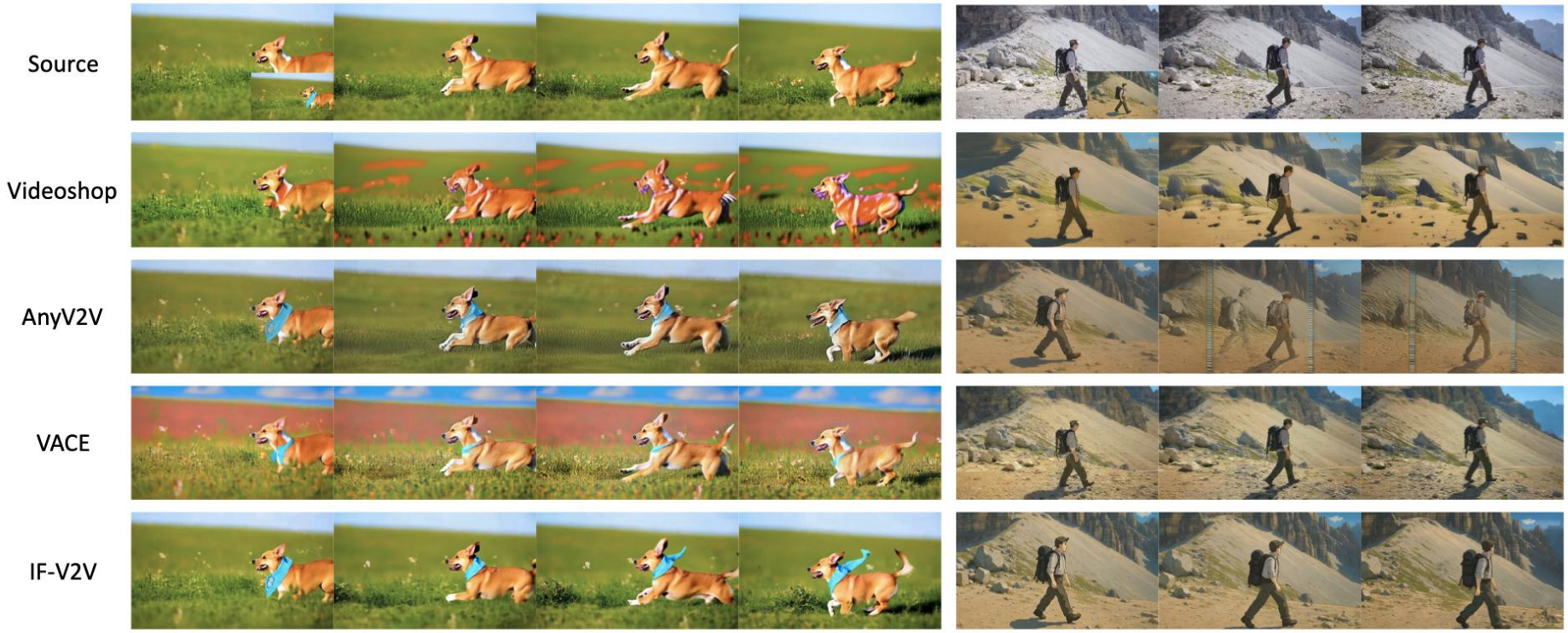
Experiments: Quantative Results

- Comparisons to prior works
  - Datasets: 40 editing samples from DAVIS and in-the-wild videos
  - Metrics:
    - Aesthetics Score (AS) : per-frame visual quality
    - Temporal Consistency (TC): video smoothness
    - Edited Frame Consistency (EFC): consistency between edited first frame and the generated video
    - Human Preferences (HP): 13 volunteers' average rating

Method	AS	TC	EFC	HP
Videoshop	4.62	97.87	76.85	1.69
AnyV2V	<u>4.81</u>	97.88	<u>81.47</u>	<u>2.56</u>
VACE†	4.57	<u>97.94</u>	75.65	1.64
IF-V2V (Ours)	**4.88**	**98.71**	**92.79**	**4.50**

Experiments: Quantative Results

- Comparisons to prior works



Experiments: Quantative Results

- Comparisons to prior works
- Ablation study

Experiments: Quantative Results

- Comparisons to prior works
- Ablation study
    - Additional metrics:
        - Original Video Consistency (OVC): per-frame consistency between the edited video and the original video
        - Average Editing Consistency (AEC): the mean value of EFC and OVC to assess the general editing consistency
        - Time: the average time taken per video for the editing process

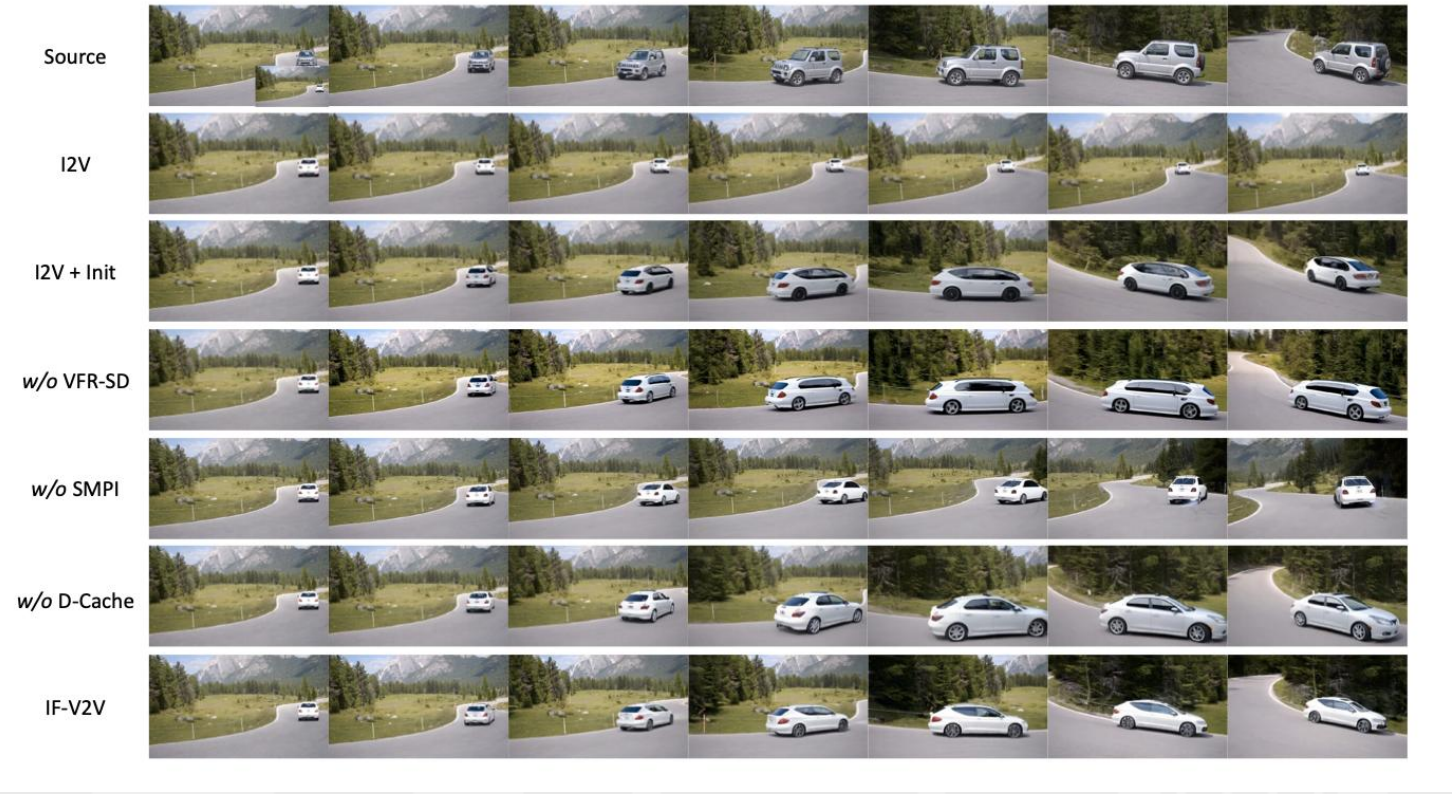
Experiments: Quantative Results

- Comparisons to prior works
- Ablation study
  - Additional metrics:
    - Original Video Consistency (OVC): per-frame consistency between the edited video and the original video
    - Average Editing Consistency (AEC): the mean value of EFC and OVC to assess the general editing consistency
    - Time: the average time taken per video for the editing process

Setting	AS	TC	EFC	OVC	AEC	Time
I2V	4.88	98.70	93.71	75.03	84.37	554.27
I2V + Init	4.89	98.30	88.34	78.74	83.54	553.52
w/o VFR-SD	4.87	98.29	91.23	75.27	83.25	**553.58**
w/o SMPI	4.78	98.19	92.67	75.45	84.06	622.38
w/o D-Cache	4.87	98.41	**93.37**	**76.61**	**84.99**	804.46
IF-V2V	**4.88**	**98.71**	92.79	76.44	84.62	616.60

Experiments: Case Study

- Creative editing sample
  - Edit first frame to “a white car with its back towards the uphill direction”



Conclusion and Discussion

- Contribution
  - Proposed IF-V2V: a user-friendly image-conditioned video editing method, leveraging strong temporal prior of pretrained flow-based I2V models
  - Equipped with Plug-and-Play module: seamless integration with updated video models
  - Versatile for diverse editing tasks: supports object addition, object removal, stylization, attribute modification, and even creative editing

Conclusion and Discussion

- Contribution
  - Proposed IF-V2V: a user-friendly image-conditioned video editing method, leveraging strong temporal prior of pretrained flow-based I2V models
  - Equipped with Plug-and-Play module: seamless integration with updated video models
  - Versatile for diverse editing tasks: supports object addition, object removal, stylization, attribute modification, and even creative editing

- Shortcomings
  - Proved to be equivalent to FlowEdit, lacking the theoratical creativity

STRUCT Group Paper Reading

THANKS FOR LISTENING!

Taming Flow-based I2V Models for Creative Video Editing | arXiv 2025

Xianghao Kong, Hansheng Chen, Yuwei Guo, Lvmin Zhang, Gordon Wetzstein, Maneesh Agrawala, Anyi Rao

Presented by Junyi Fan 2025.11.23