Diffusion Models for Visual Computing

Chun-Hao Huang

Niloy Mitra, Daniel Cohen-Or, Minhyuk Sung, Duygu Ceylan, Paul Guerrero

Part 5: Beyond Single Images

https://geometry.cs.ucl.ac.uk/courses/diffusion4VC_eg24/

Presentation Schedule

Introduction to Diffusion Models

Guidance and Conditioning Sampling

Attention

Break

Personalization and Editing

Beyond Single Images

Diffusion Models for 3D Generation

So Far

- Theory and principles of diffusion models
- 2D image generation and editing
  - Editing one single image
  - Identity preservation

This Section

- Editing “multiple images”
  - in temporal dimension → videos
  - in 2D image domain → image montage
  - in 3D spatial domain → multi-views (*without* explicit 3D awareness)

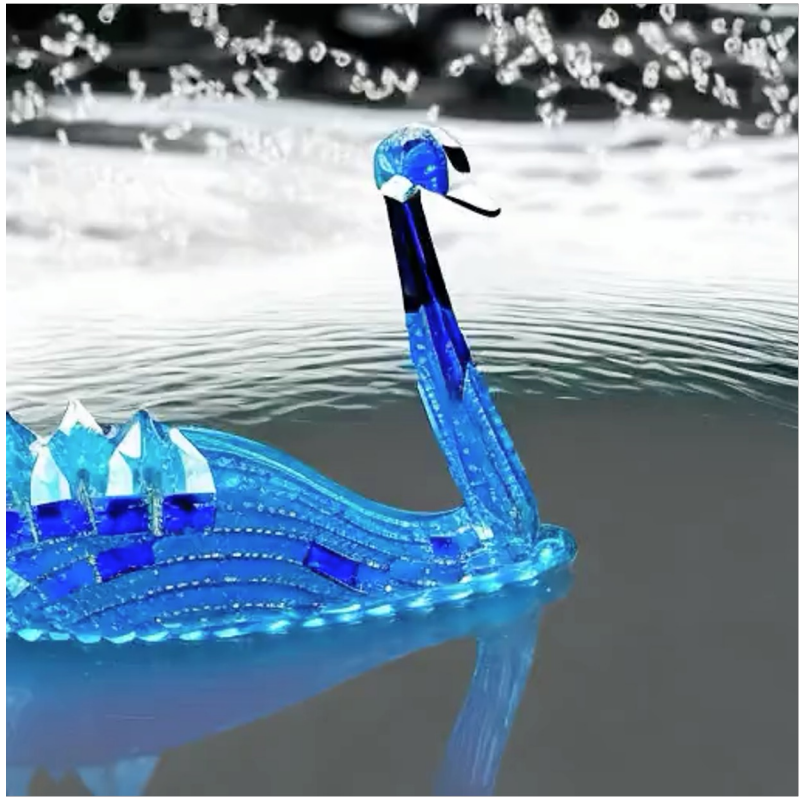
- Training-free / zero-shot setup
  - Repurposing existing pretrained image diffusion model

- Training setup
  - Video diffusion model

Individual Editing – Videos

input

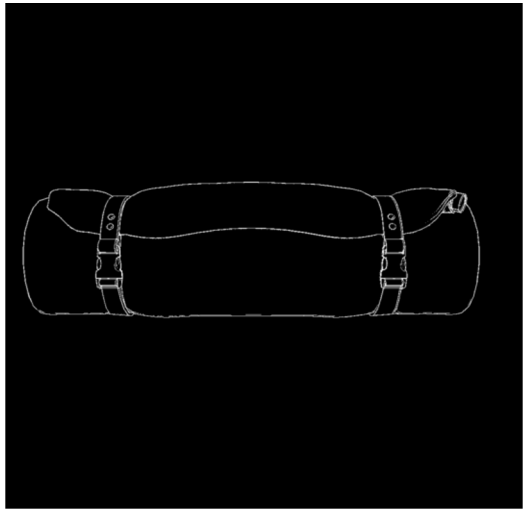
per-frame edit



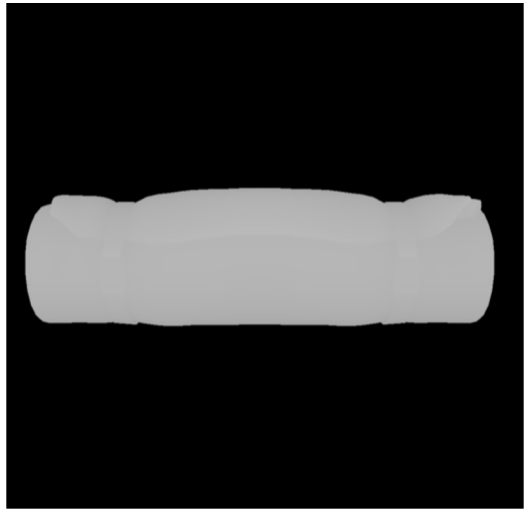
depth-conditioned SD

Individual Editing – Multiple Views

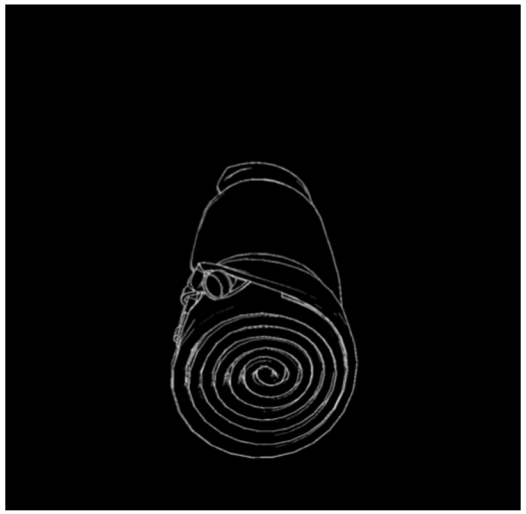
input

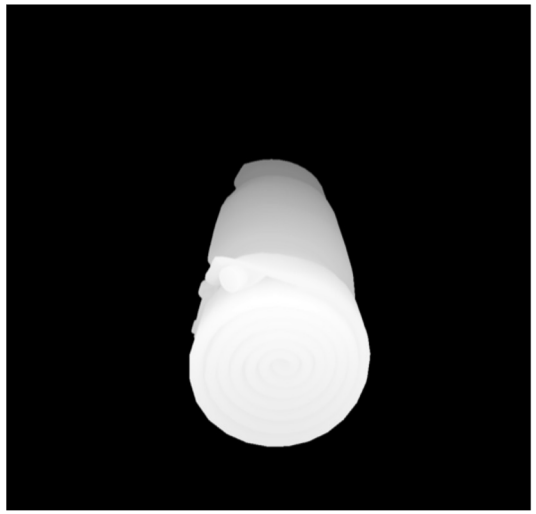


input



per-view edits





Firefly structure match

Design Space for Consistency

- If *correspondences are known*, synchronzing:
  1. Initial noise and/or noisy latents
  2. Features
- Loss-guided denoising.
- Repurposing self attention for cross-view, cross-frame attention.

Noise Model

- Denoting the initial noise of each frame as $\epsilon^1, \epsilon^2, \dots, \epsilon^i, \dots$
- I.I.D.: $\epsilon^1, \epsilon^2, \dots, \epsilon^i, \dots \sim \mathcal{N}(0, \mathbf{I})$
- Re-using the same noise: $\epsilon^1 = \epsilon^2 = \epsilon^i = \cdots \sim \mathcal{N}(0, \mathbf{I})$
- Mixed Noise Model [1]

$$\epsilon_{\text{shared}} \sim \mathcal{N}\left(0, \frac{\alpha^2}{1+\alpha^2}\mathbf{I}\right), \epsilon_{\text{ind}}^i \sim \mathcal{N}\left(0, \frac{\alpha^2}{1+\alpha^2}\mathbf{I}\right)$$
$$\epsilon^i = \epsilon_{\text{shared}} + \epsilon_{\text{ind}}^i$$

- Progressive Noise Model [1]

$$\epsilon^0 \sim \mathcal{N}(0, \mathbf{I}), \epsilon_{\text{ind}}^i \sim \mathcal{N}\left(0, \frac{1}{\sqrt{1+\alpha^2}}\mathbf{I}\right)$$
$$\epsilon^i = \frac{\alpha}{1+\alpha^2}\epsilon^{i-1} + \epsilon_{\text{ind}}^i$$

[1] Ge et al., Preserve Your Own Correlation: A Noise Prior for Video Diffusion Models, *ICCV'23*

Correspondence-guided Noise

Kass and Pesare, Coherent noise for non-photorealistic rendering, *ToG'11*

Correspondence-guided Noise

- $\epsilon^0 = \sqrt{\frac{1-\alpha}{1+\alpha}}\epsilon^0_{\text{ind}}$, where $\epsilon^0_{\text{ind}} \sim \mathcal{N}(0, \mathbf{I})$

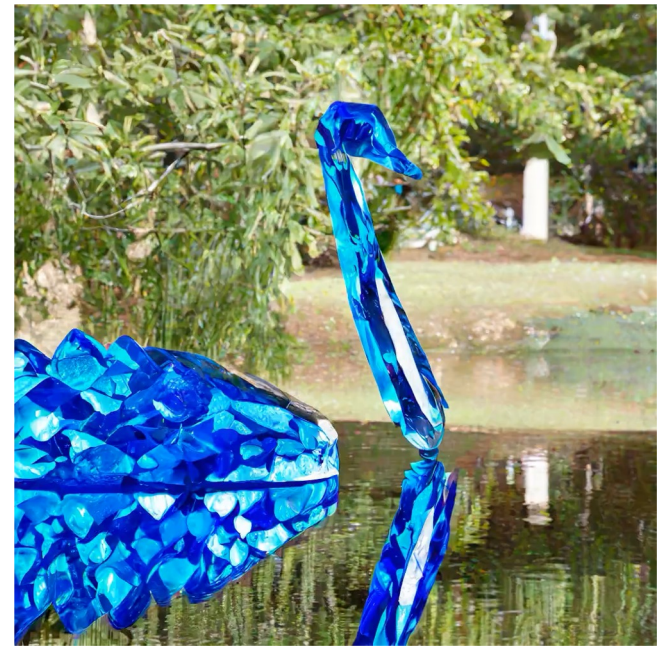
- $$\epsilon^i(x,y) = \begin{cases} \sqrt{\frac{1-\alpha}{1+\alpha}}\epsilon^i_{\text{ind}}(x,y), & \epsilon^i_{\text{ind}} \sim \mathcal{N}(0, \mathbf{I}) \\ \alpha\epsilon^{i-1}(x',y') + (1-\alpha)\epsilon^i_{\text{ind}}(x,y) \end{cases}$$

disocclusion

if correspondence pairs (x', y') & (x, y) exist e.g., obtained by optical flow.

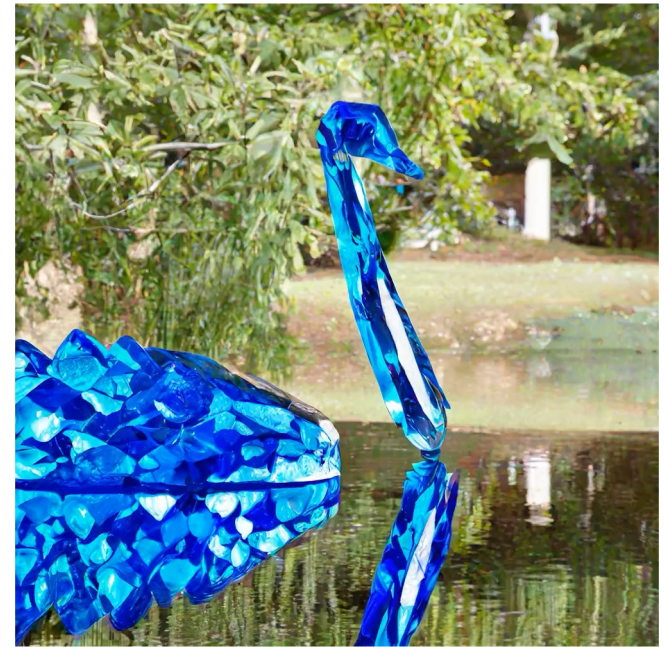
input

results using same rand initialized noise



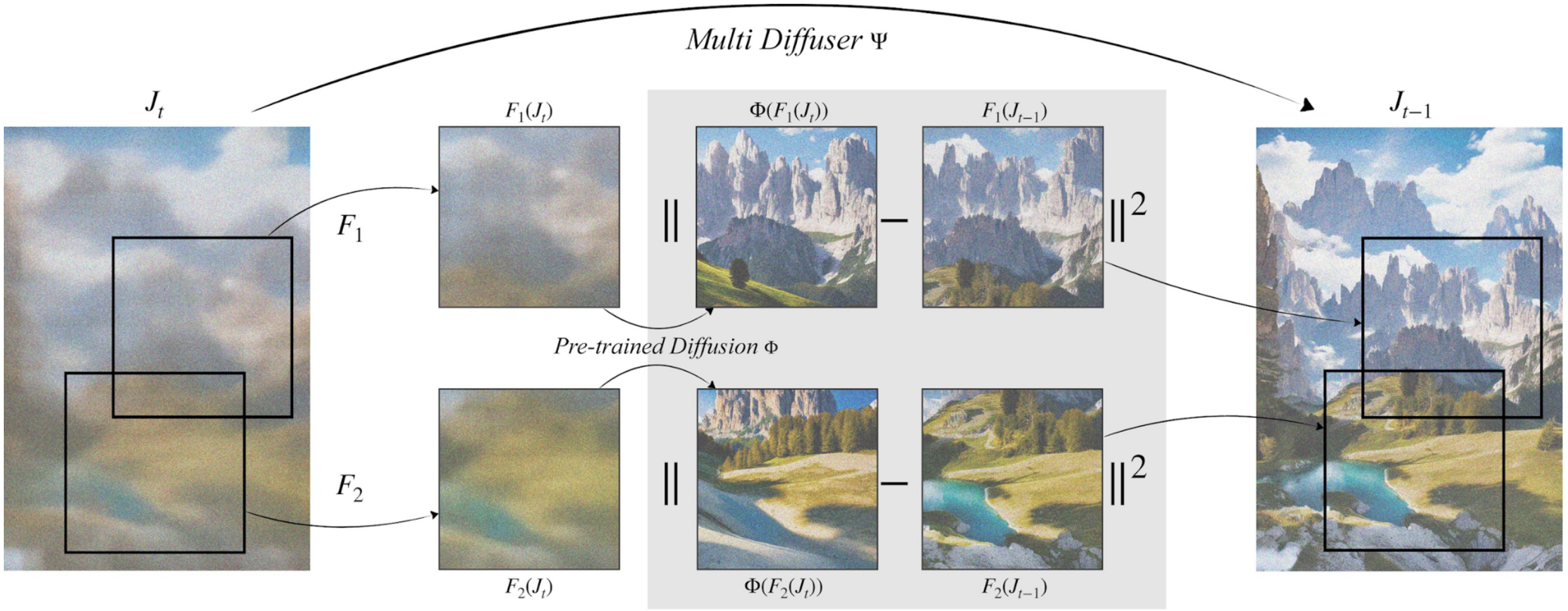
correspondence guided noise

results using correspondence guided noise



Synchronizing Colors or Latent Features

- Averaging noisy latents of the same pixels



Bar-Tal et al., MultiDiffusion: Fusing Diffusion Paths for Controlled Image Generation, *ICML'23*

Loss-guided Denoising

$\mathcal{D}(\mathbf{x}_t)$ noisy image at t=45

0.6485 0.6433

$\mathcal{D}(\phi_\theta(\mathbf{x}_t, t))$ predicted x0 at t=45

0.5416 0.3501

$\mathcal{D}(\mathbf{x}_0)$ final denoised x0

0.5911 0.3704

$$\hat{\mathbf{x}}_t^{(i)} = \mathbf{x}_t^{(i)} - w\nabla_{\mathbf{x}_t^{(i)}}\mathcal{L}\left(\mathcal{D}(\phi_\theta(\mathbf{x}_t^{(i)}, t)), \mathcal{D}(\phi_\theta(\mathbf{x}_t^{(0)}, t))\right)$$

Loss $\mathcal{L}$: LPIPS score on the predicted denoised x0

Lee et al., SyncDiffusion: Coherent Montage via Synchronized Joint Diffusions, *NeurIPS'23*

without loss-guided denoising

with loss-guided denoising

MultiDiffusion

Fusing Diffusion Paths for Controlled Image Generation
(averaging noisy latents)

Code

https://github.com/omerbt/MultiDiffusion

SyncDiffusion

Coherent Montage via Synchronized Joint Diffusions
(loss-guided denoising)

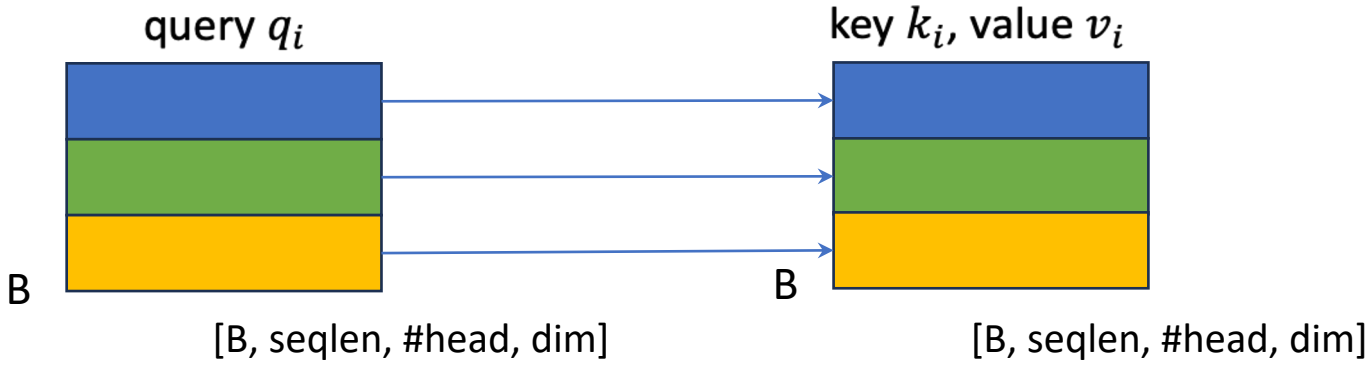
Code

https://github.com/KAIST-Visual-AI-Group/SyncDiffusion

Self-attention in UNet

$$Att(q_i, k_i, v_i) = softmax\left(\frac{q_i k_i^{\mathrm{T}}}{\sqrt{d}}\right) v_i$$

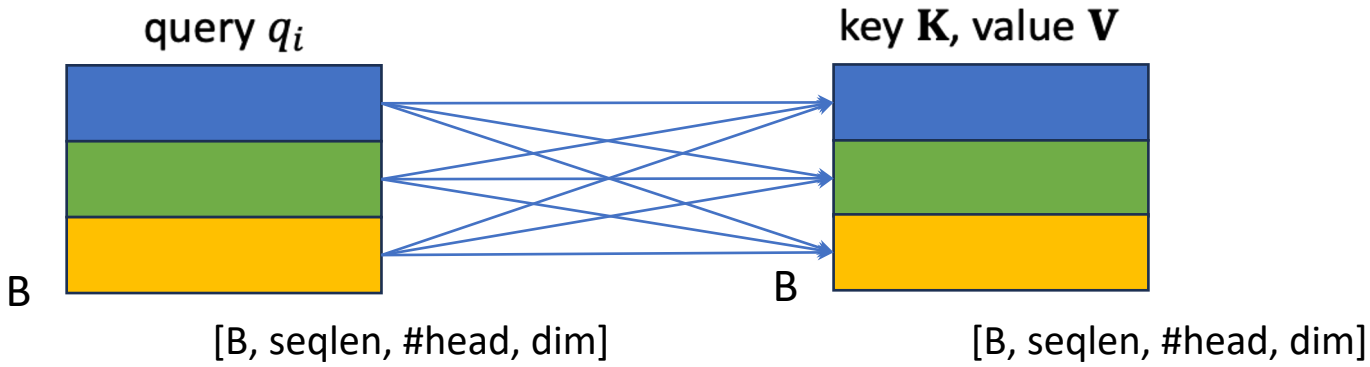
- Independent generation: each sample in the batch attends to only itself.



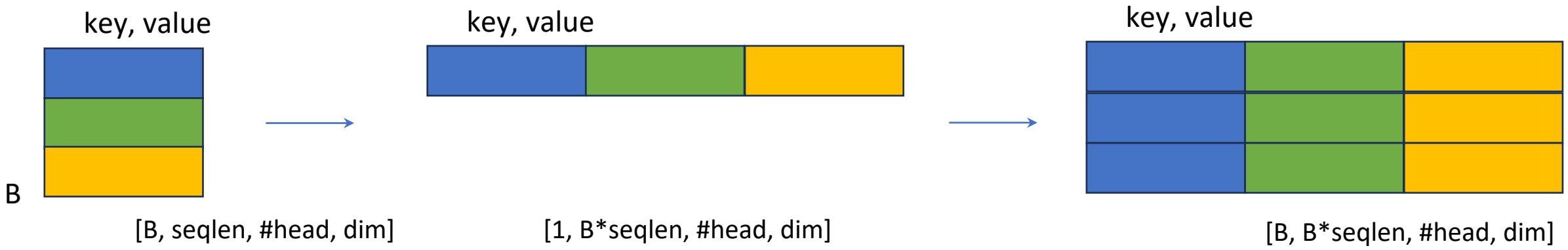
Repurposing Self Attention

$$Att(q_i, \mathbf{K}, \mathbf{V}) = softmax\left(\frac{q_i \mathbf{K}^T}{\sqrt{d}}\right)\mathbf{V},$$
$$\text{where } \mathbf{K} = [k_1, \dots k_i, \dots], \mathbf{V} = [v_1, \dots v_i, \dots]$$

- Batch generation: each sample attends to every samples.

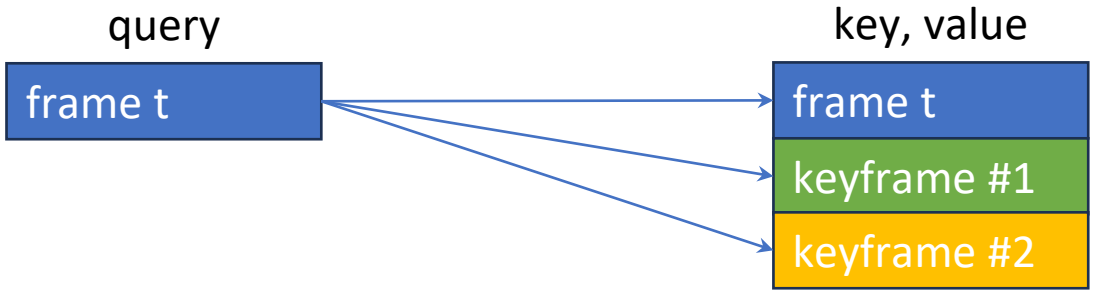


Pseudo Code




```
bs = k.shape[0]
k, v = (t.reshape(1, -1, attn.heads,attn.dim_head) for t in (k, v))
k = k.expand(bs, -1, -1, -1)
v = v.expand(bs, -1, -1, -1)
```

Training-free/zero-shot Video Stylization



- Each frame attends to pre-defined “keyframes”
  - Pix2Video [1]: keyframes = [frame 1, frame t-1]
  - FateZero [2]: keyframes = [middle frame]
  - Text2Video-Zero [3]: keyframes = [frame 1]

[1] Ceylan et al., Pix2Video: Video Editing using Image Diffusion, *ICCV’23*
[2] Qi et al., FateZero: Fusing Attentions for Zero-shot Text-based Video Editing, *ICCV’23*
[3] Khachatryan et al., Text2Video-Zero: Text-to-Image Diffusion Models are Zero-Shot Video Generators, *ICCV’23*

Text-guided Video Stylization

a group of chocolate pigs looking for food

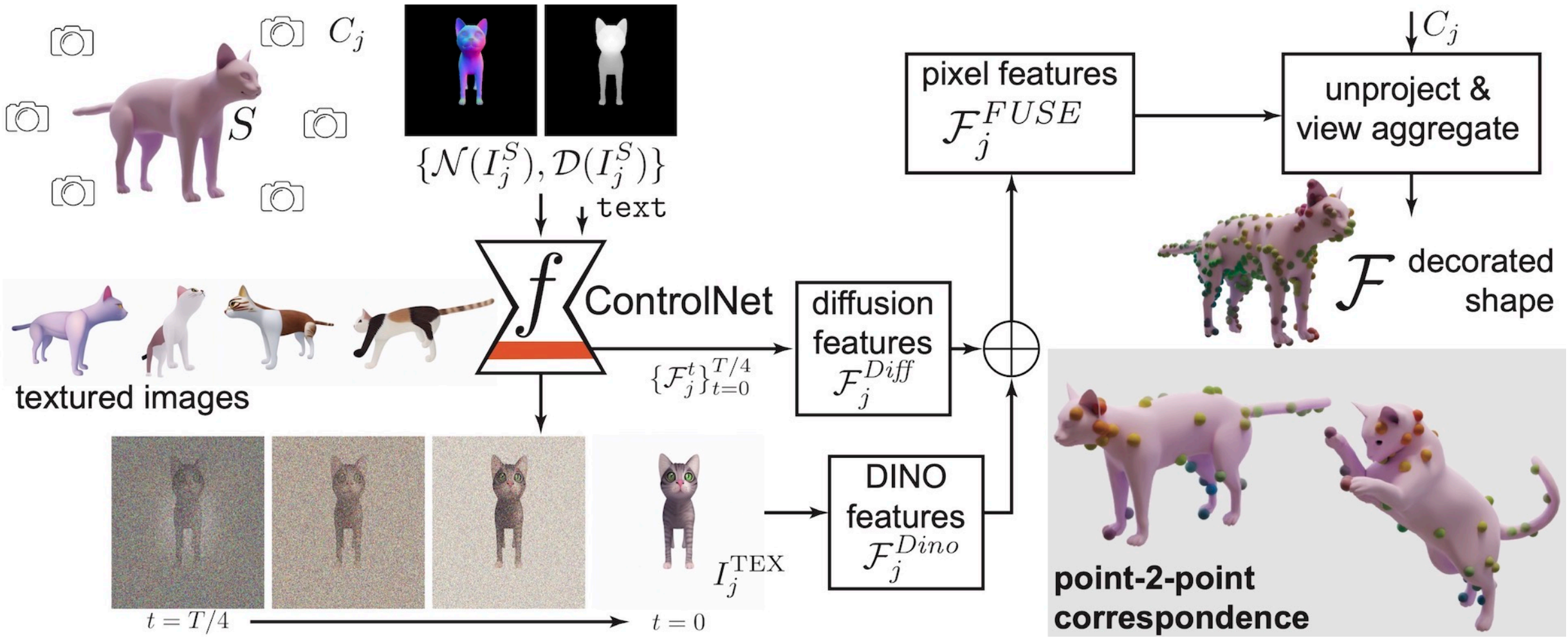
Ceylan et al., Pix2Video: Video Editing using Image Diffusion, *ICCV'23*

Another Example

a Swarovski blue crystal swan on the lake

Ceylan et al., Pix2Video: Video Editing using Image Diffusion, *ICCV'23*

Diffusion Features



Dutt et al., Diffusion 3D Features (Diff3F): Decorating Untextured Shapes with Distilled Semantic Features, *CVPR'24*

Diffusion Features

Dutt et al., Diffusion 3D Features (Diff3F): Decorating Untextured Shapes with Distilled Semantic Features, *CVPR'24*

Pix2Video

Video Editing using Image Diffusion

Code

https://github.com/duyguceylan/pix2video

FateZero

Fusing Attentions for Zero-shot Text-based Video Editing

Code

https://github.com/ChenyangQiQi/FateZero

Text2Video-Zero

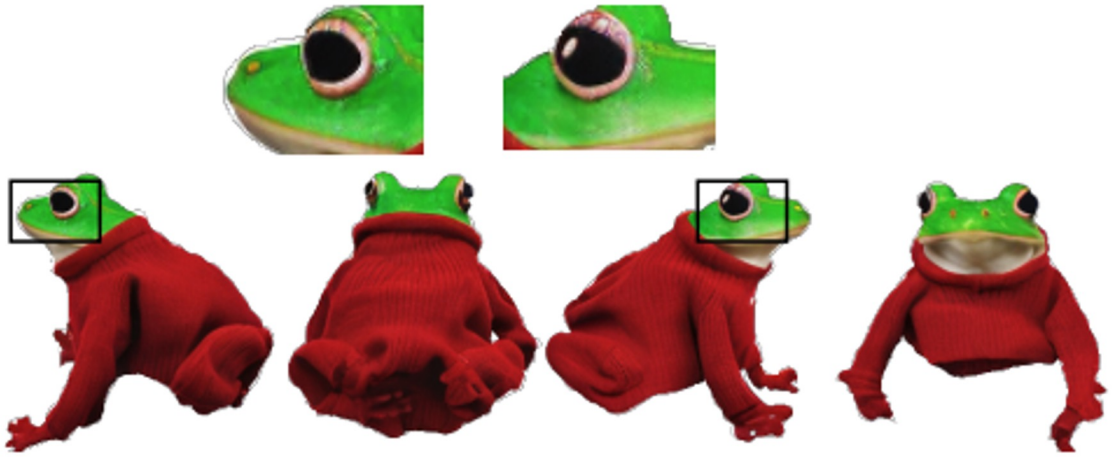
Text-to-Image Diffusion Models are Zero-Shot Video Generators

Code

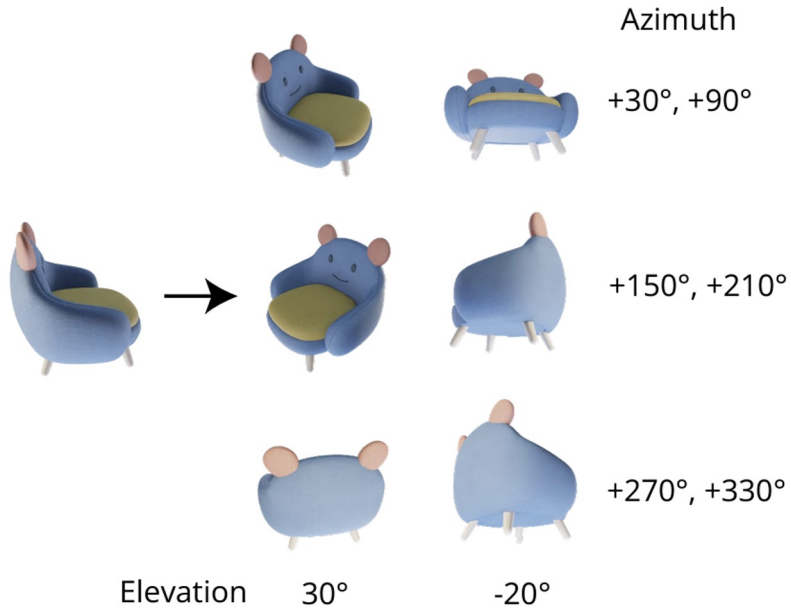
https://github.com/Picsart-AI-Research/Text2Video-Zero

Applied on Multi-view Generation

- Stacking views into an image grid
  a simple yet effective way of repurposing self-attention as cross-view attention.



Tsalicoglou et al., Textmesh: Generation of realistic 3D meshes from text prompts, arXiv'23



Shi et al., Zero123++: a Single Image to Consistent Multi-view Diffusion Base Model, arXiv'23

Zero123++

A Single Image to Consistent Multi-view Diffusion Base Model
(stacking 6 predefined views)

Code

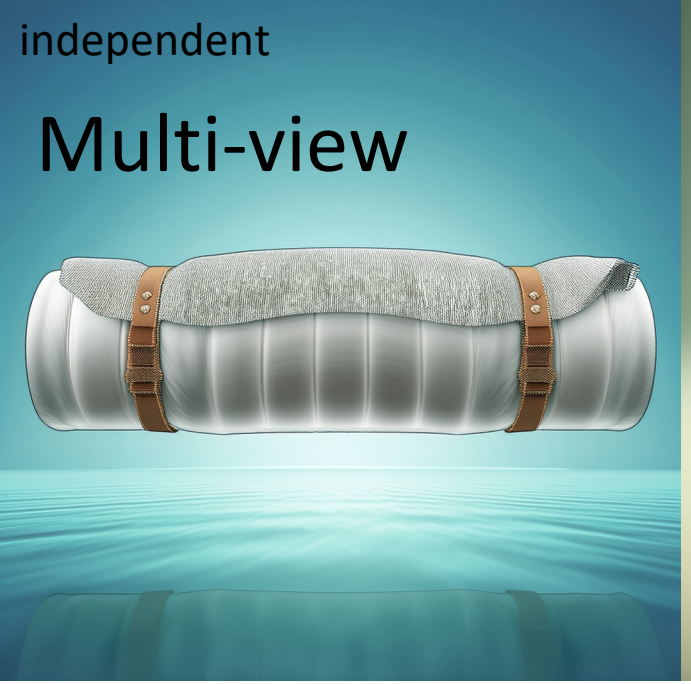
https://github.com/SUDO-AI-3D/zero123plus

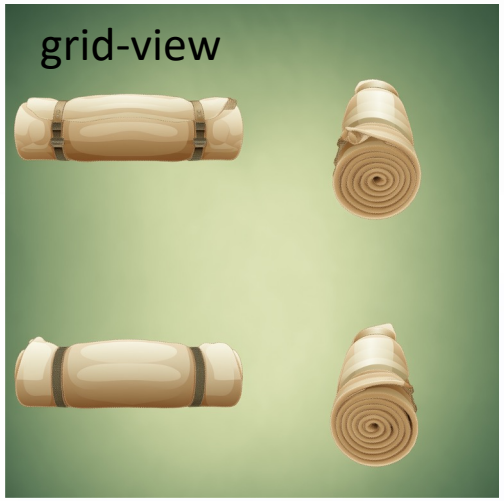
Demo

https://github.com/SUDO-AI-3D/zero123plus

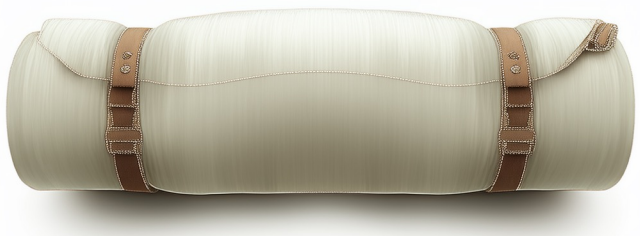
independent

Multi-view



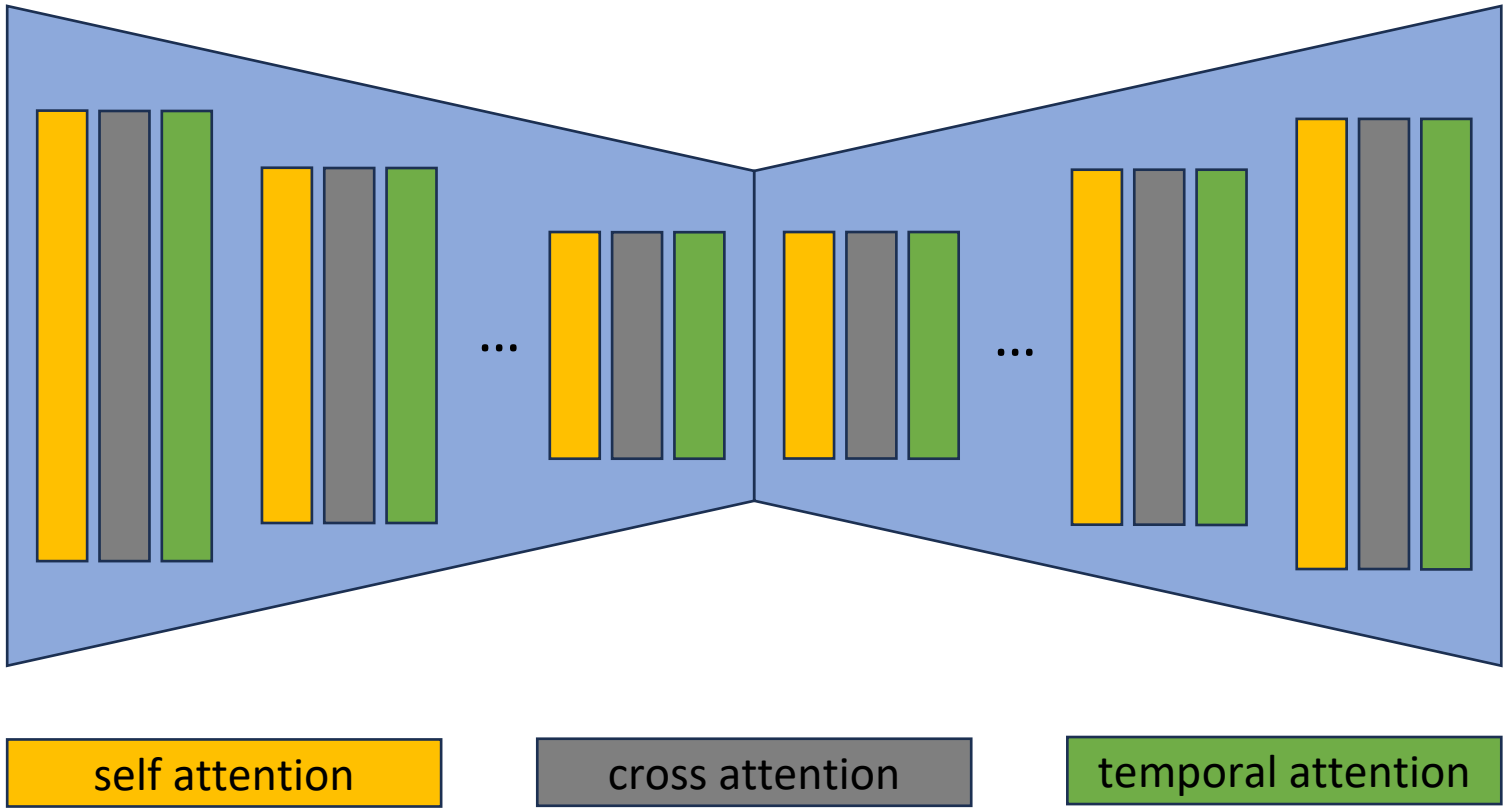


batch

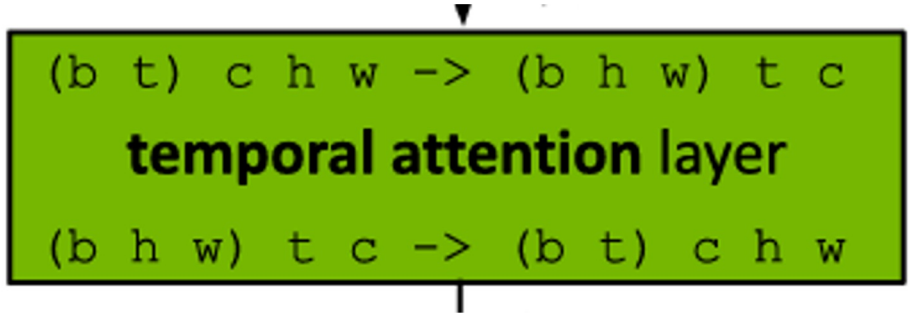


Video Diffusion Model

- Adding a temporal attention layer after spatial cross attention



Inflated Temporal Attention Layer



Blattmann et al., Align your Latents: High-Resolution Video Synthesis with Latent Diffusion Models, *CVPR'23*

Blattmann et al., Stable Video Diffusion: Scaling Latent Video Diffusion Models to Large Datasets, *arXiv'23*

Guo et al., AnimateDiff: Animate Your Personalized Text-to-Image Diffusion Models without Specific Tuning, *ICLR'24*

Code

https://github.com/huggingface/diffusers/blob/main/src/diffusers/models/transformers/transformer_temporal.py

Applied on Multi-view Generation

Generating 360 turn-table videos of 3D objects

Blattmann et al., Stable Video Diffusion: Scaling Latent Video Diffusion Models to Large Datasets, arXiv'23