Qualcomm

Jan 25, 2024

# Efficient generative AI for images and video

**Amirhossein Habibian**

Director of Engineering
Qualcomm Technologies Netherlands

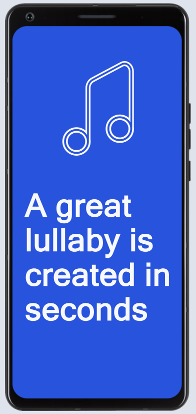
# Agenda

- The potential impact of efficient generative vision
- Efficient image generation
- Efficient video generation
- Efficient 3D generation
- Applications: automotive
- Q&A

## Text generation
(ChatGPT, Bard, Llama, etc.)

Input prompts

"Write a lullaby about cats and dogs to help a child fall asleep, include a golden shepherd"




### Real-life application

- Communications
- Journalism
- Publishing
- Creative writing
- Writing assistance

## Image generation
(Stable Diffusion, MidJourney, etc.)

Input prompts

"Super cute fluffy cat warrior in armor"

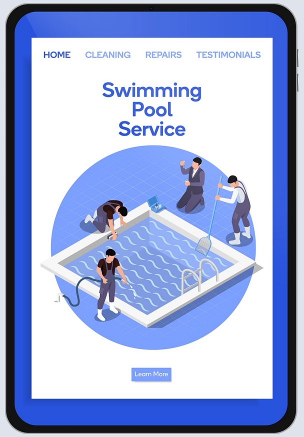
### Real-life application

- Advertisements
- Published illustrations
- Corporate visuals
- Novel image generation

## Code generation
(Codex, etc.)

Input prompts

"Create code for a pool cleaning website with tab for cleaning, repairs, and testimonials"




A beautiful website is created in seconds

### Real-life application

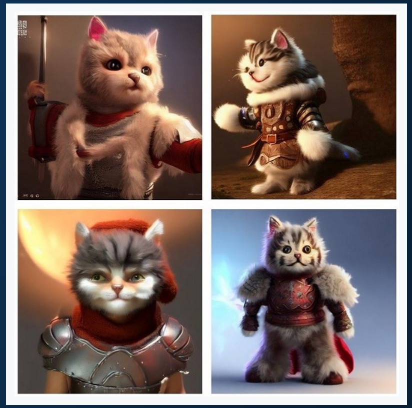
- Web design
- Software development
- Coding
- Technology

# What is generative AI?

AI models that create new and original content like text, images, video, audio, or other data

Generative AI, foundational models, and large language models are sometimes used interchangeably

# Why is generative AI for computer vision important?



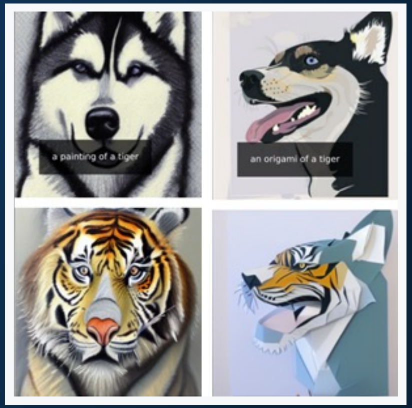
## Generating images and videos

Generative models create images and videos from scratch

Original, life-like visuals generated from textual and/or image prompts (in the case of image/text-to-image or image/text-to-video)

**Examples:**
Stable Diffusion, ControlNet

Prompt: “Super cute fluffy cat warrior in armor, photorealistic, 4K, ultra detailed, vray rendering, unreal engine”




## Editing images and videos

Generative models change aspect of images and videos

Swap the background, change style or edit object’s attribute and appearance

**Examples:**
SDEdit, PnP, Pix2Pix

## Generating 3D content

Generative models create 3D meshes and assets

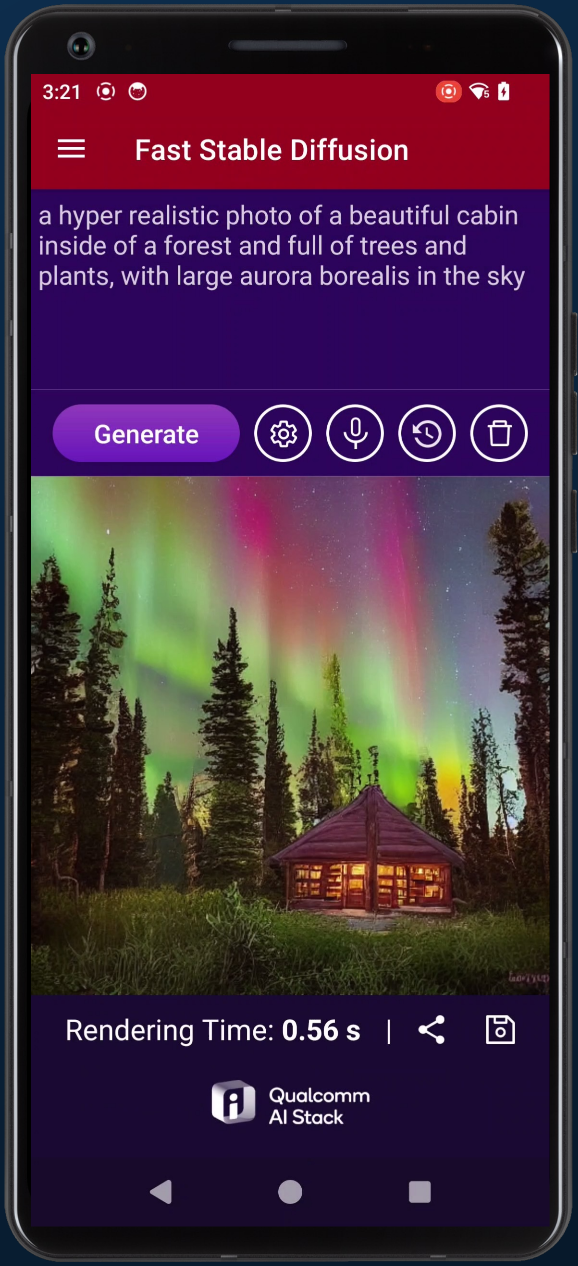
Based on textual description or a handful of images with minimal manual effort

**Examples:**
DreamFusion, Magic3D

Prompt: “a plush dragon toy”

AT SNAPDRAGON SUMMIT 2023

# World's fastest AI text-to-image generative AI on a phone




## Fast Stable Diffusion

Takes less than 0.6 seconds for generating 512x512 images from text prompts

Efficient UNet architecture, guidance conditioning, and step distillation

Full-stack AI optimization to achieve this improvement

# What are the challenges to overcome for generative AI images and videos?

## High computation and latency

Gen AI requires immense computational power and infrastructure.

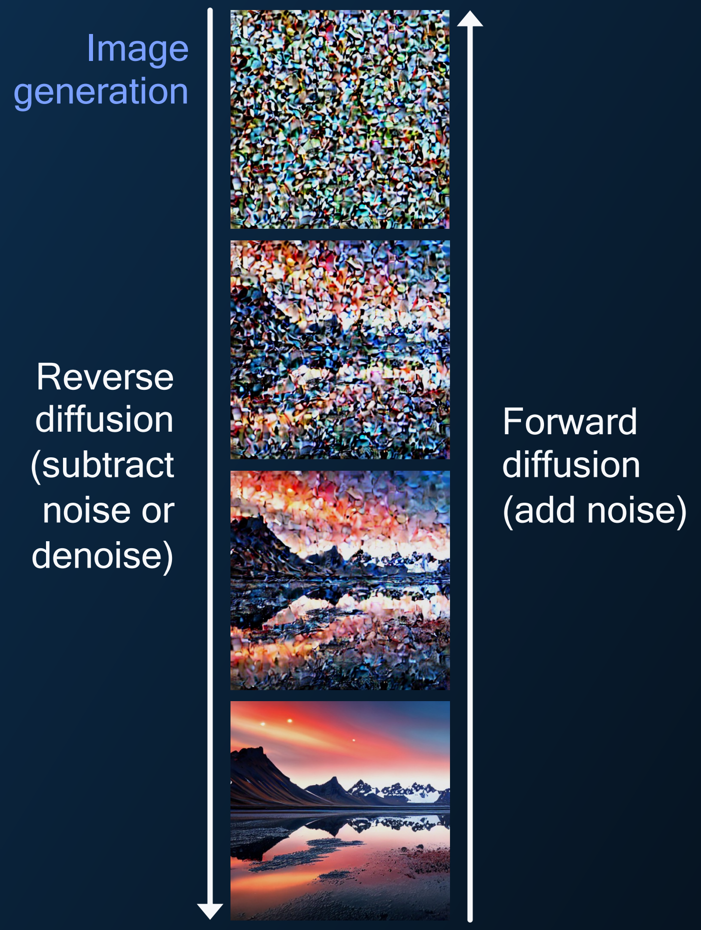
## Memory costs

Models demand a lot of memory to perform well and sometimes need to run concurrently.

## Data inefficiency

Models require billions of training samples, that makes it hard to adapt them to new domains.

# What is diffusion?



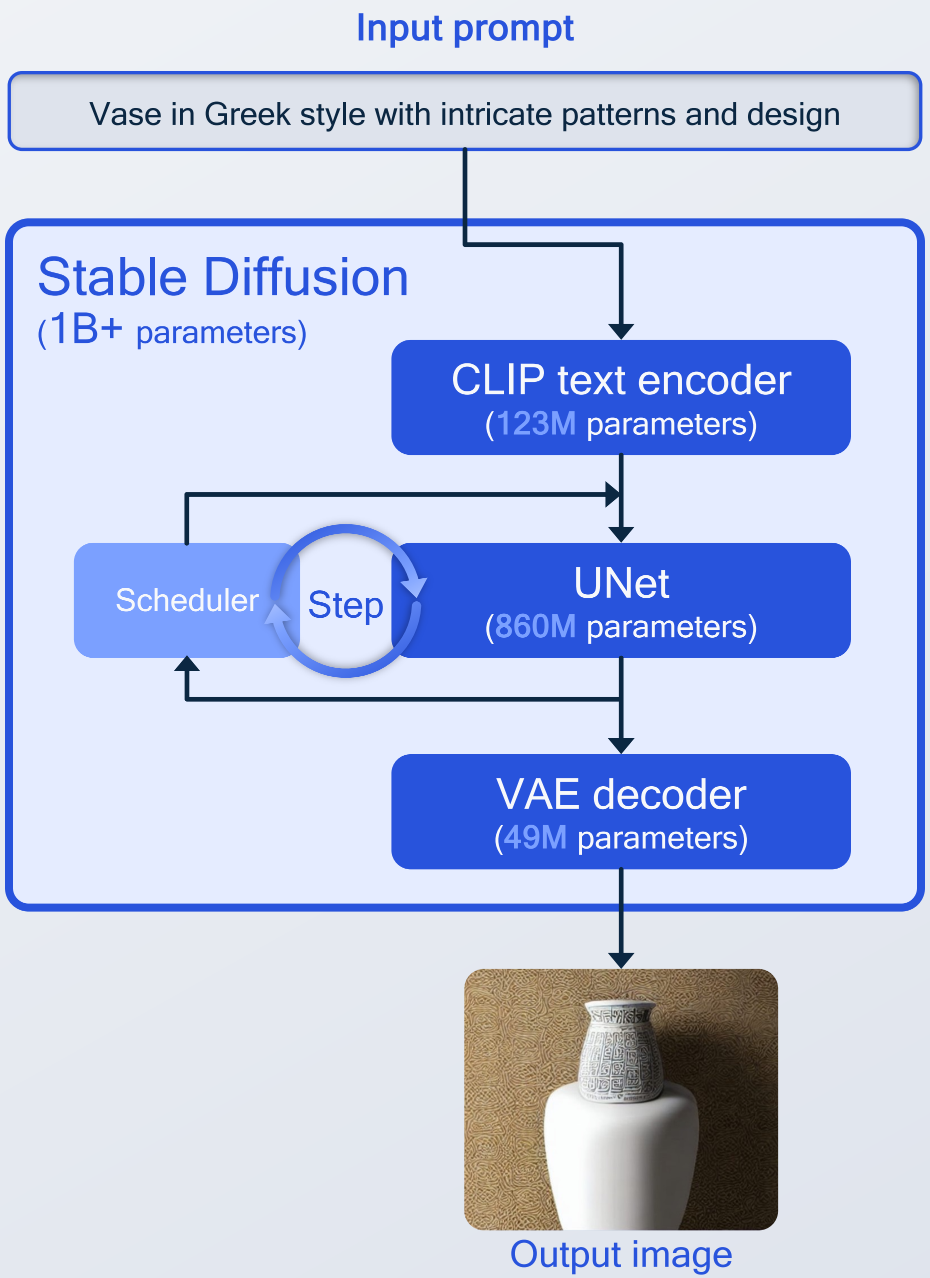

Prompt: Panoramic view of mountains of Vestrahorn and perfect reflection in shallow water, soon after sunrise, Stokksnes, South Iceland, Polar Regions, natural lighting, cinematic wallpaper

# Stable Diffusion architecture

UNet is the biggest component model of Stable Diffusion

Many steps, often 20 or more, are used for generating high-quality images

Significant compute is required

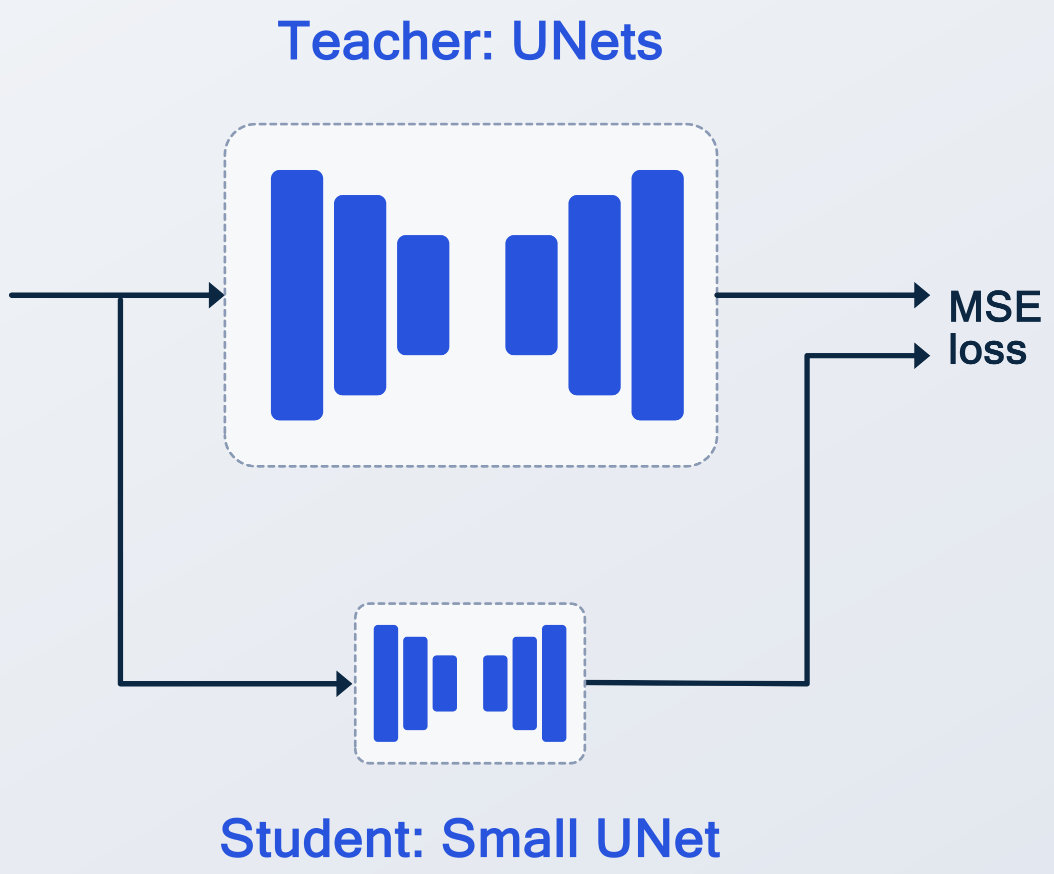



VAE: Variational Auto Encoder;
CLIP: Contrastive Language-Image Pre-Training

Key concept:

# Model distillation

Teach the student model to achieve what the teacher achieves at each step

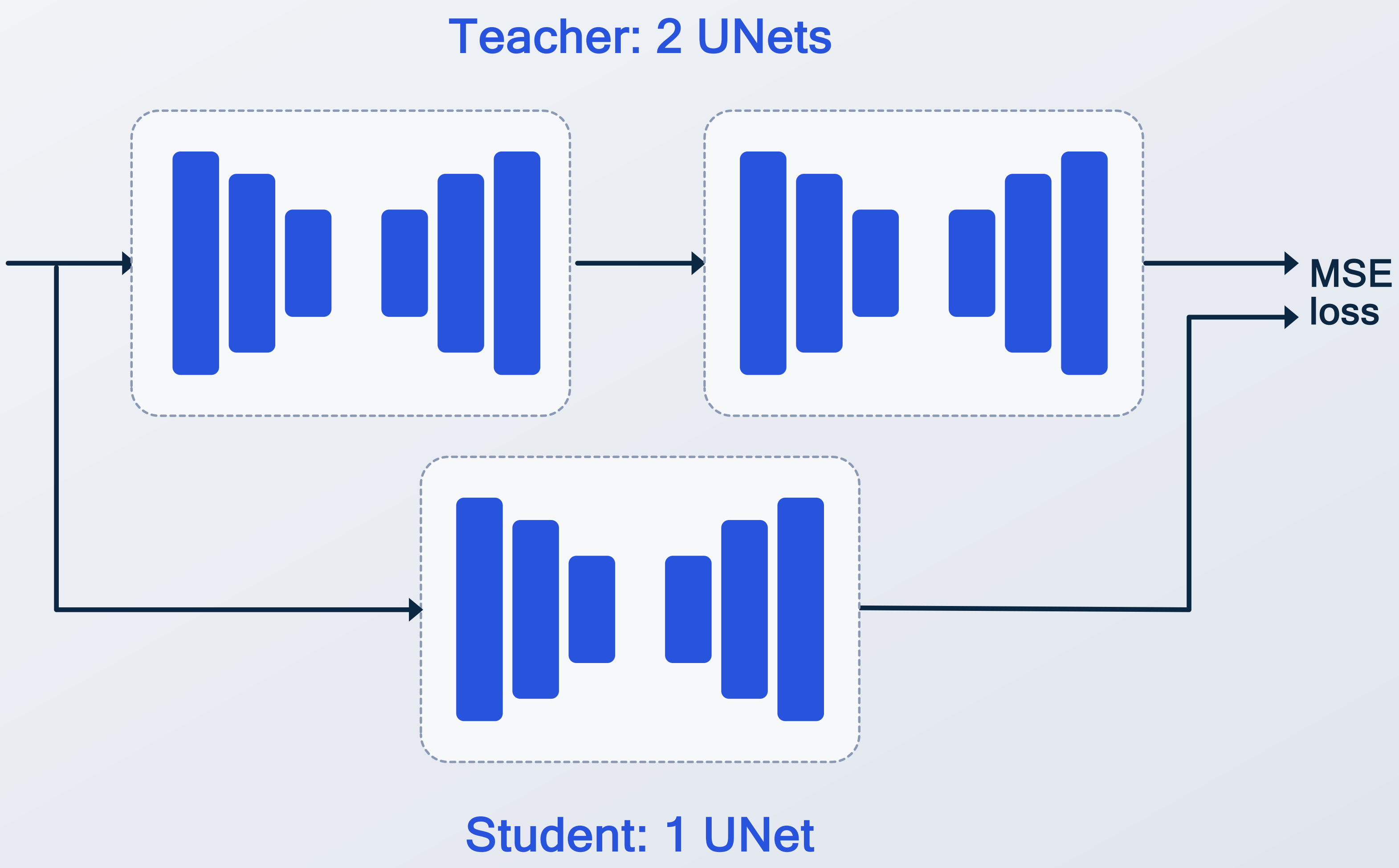



BK-SDM: Architecturally Compressed Stable Diffusion for Efficient Text-to-Image Generation, arXiv'23

## Key concept:

# Step distillation

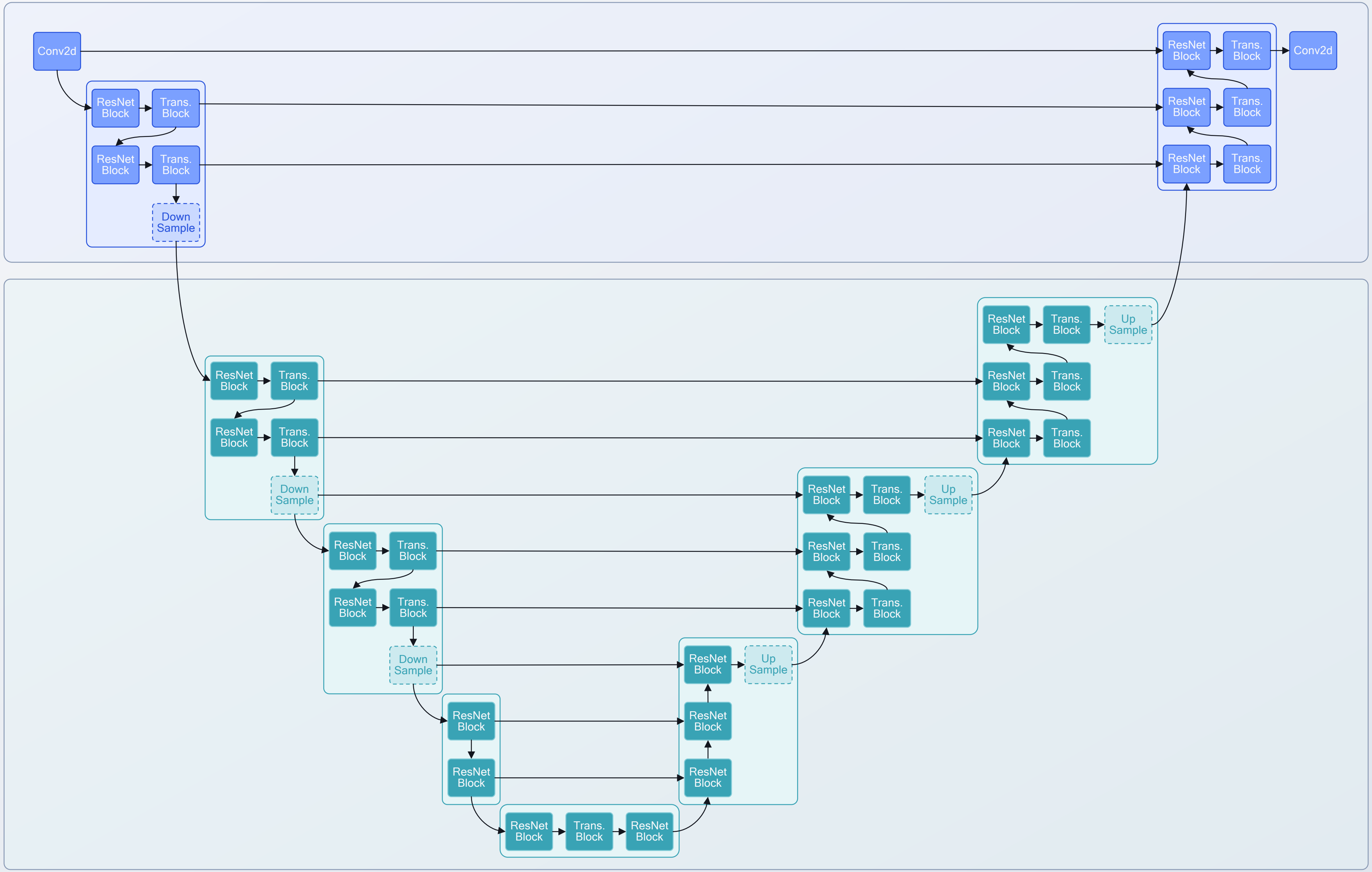
Teach the student model to achieve in one step what the teacher achieves in multiple steps

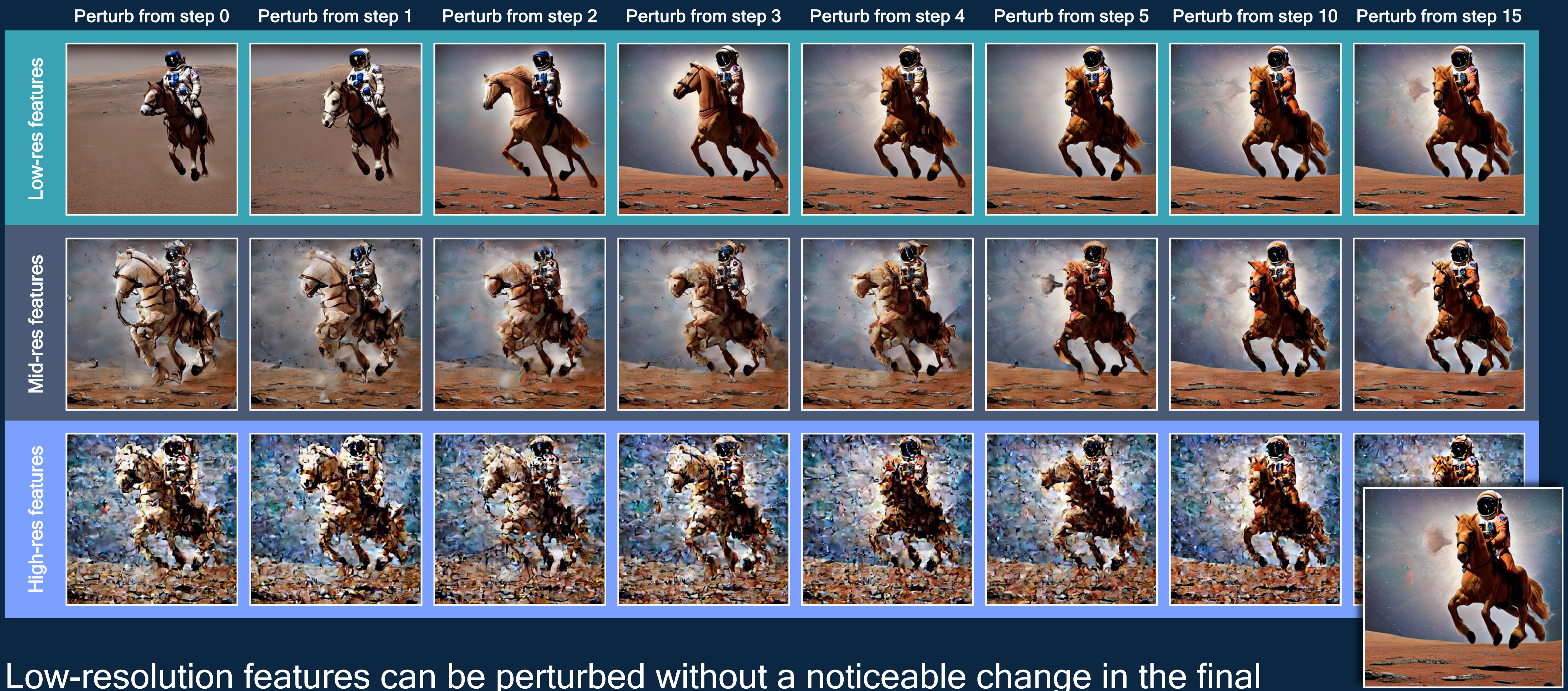



Progressive Distillation for Fast Sampling of Diffusion Models, ICLR'22

High-resolution representations in UNet carry high-frequency content (e.g., textures)

Low-resolution representations in UNet carry high-level structure (e.g., scene layout)



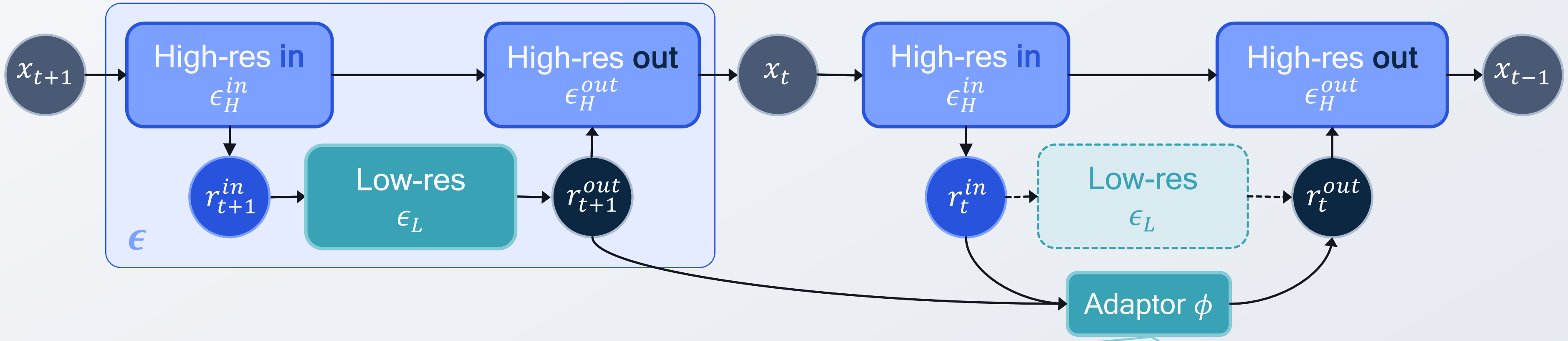



Low-resolution features can be perturbed without a noticeable change in the final output, whereas small perturbations on the high-resolution features degrade the image generation

Prompt: "image of an astronaut riding a horse on mars"

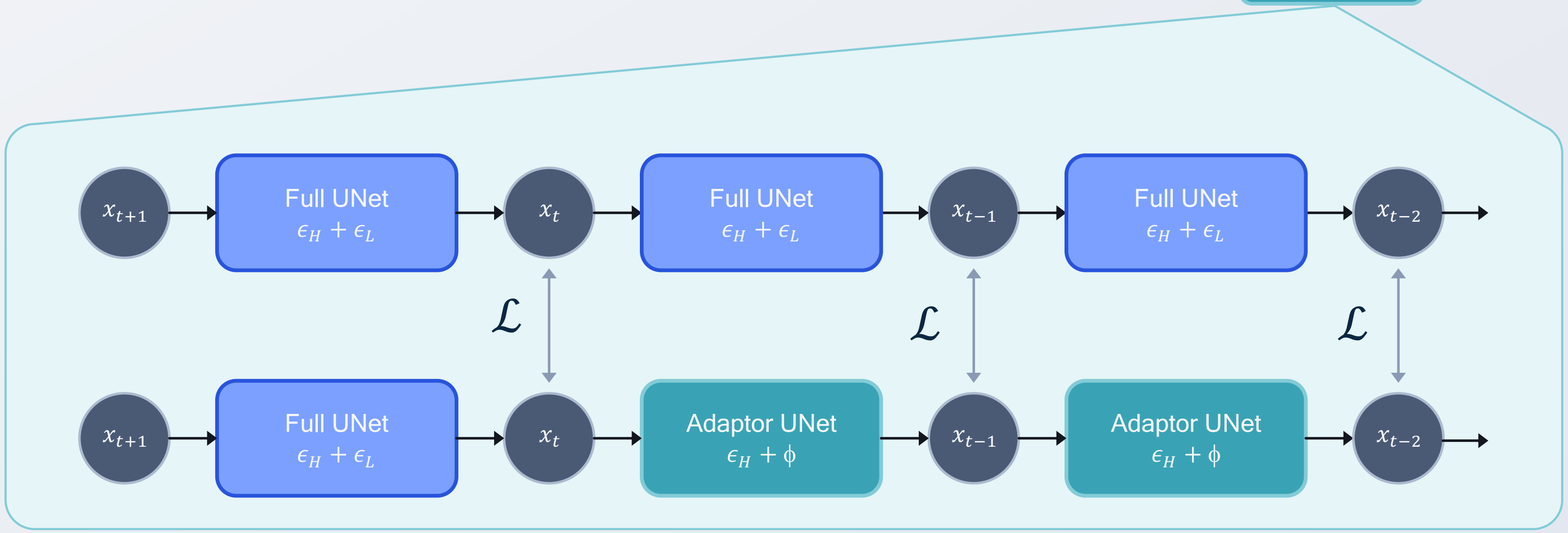
## Clockwork architecture

An efficient approximation of low-res features by adapting from previous steps




## Training the adaptor

Distillation from a full UNet over all denoising steps




How to leverage the perturbation robustness to save computations?

# Clockwork improves any diffusion model

## Text-to-image generation on MS-COCO 2017-5K

| Model | FID ↓ | CLIP ↑ | FLOPs $(10^{12})$ ↓ |
|---|---|---|---|
| Stable Diffusion UNet | 24.64 | 0.300 | 10.8 |
| + Clockwork | 24.11 | 0.295 | 7.3 (**1.48**×) |
| Efficient UNet | 24.22 | 0.302 | 9.5 |
| + Clockwork | 23.21 | 0.296 | 5.9 (**1.61**×) |
| Distilled Efficient UNet | 25.75 | 0.297 | 4.7 |
| + Clockwork | 24.45 | 0.295 | 2.9 (**1.62**×) |

FID = Frechet Inception Distance, CLIP = Contrastive Language-Image Pre-training

# Clockwork generates high-quality images faster than state of the art

## Text-to-image generation on MS-COCO 2017-5K

| Model | FID ↓ | CLIP ↑ | FLOPs $(10^{12})$ ↓ |
|---|---|---|---|
| InstalFlow (1 step)[1] | 29.30 | 0.283 | 0.8 |
| Model Distillation[2] | 31.48 | 0.268 | 7.8 |
| Guidance Distillation[3] | 26.90 | 0.300 | 6.4 |
| SnapFusion[4] | 24.20 | 0.300 | 4.0 |
| **Clockwork** | **24.45** | **0.295** | **2.9** |

[1] Instaflow: One step is enough for high-quality diffusion-based text-to-image generation. arXiv'2
[2] On architectural compression of text-to-image diffusion models, arXiv'23
[3] On distillation of guided diffusion models, CVPR'23
[4] SnapFusion: Text-to-image diffusion model on mobile devices within two seconds, NeurIPS'23

FID = Frechet Inception Distance, CLIP = Contrastive Language-Image Pre-training

Baseline Stable Diffusion

1. **e-to-v** — Reparameterization from epsilon to velocity space for robust distillation
2. **Efficient UNet** — Reduces compute (FLOPs), model size, peak memory usage
3. **Guidance conditioning** — Combines conditional and unconditional generation
4. **Step distillation** — Reduces UNet forward passes to less than 20

Fast Stable Diffusion

5. **Clockwork distillation** — Reduce cost of UNet for some forward passes

Faster Stable Diffusion

| | FID↓ | CLIP ↑ | Diffusion latency | Total latency |
|---|---|---|---|---|
| Fast Stable Diffusion | 26.04 | 0.297 | 0.40 seconds | 0.65 seconds |
| Faster Stable Diffusion | 25.21 | 0.292 | 0.27 seconds | 0.53 seconds |

**1.2x** Speedup

## Clockwork reduces the total latency by 1.2x while improving quality compared to Fast Stable Diffusion

# Results: state-of-the-art efficient image generation by Clockwork

Stable Diffusion v1.5

Stable Diffusion v1.5 +Clockwork

Fast Stable Diffusion

Fast Stable Diffusion +Clockwork

Prompts: "large white bear standing near a rock", "the vegetables are cooking in the skillet on the stove.", "bright kitchen with tulips on the table and plants by the window ", "red clouds as sun sets over the ocean", "a picnic table with pizza on two trays ", "a couple of sandwich slices with lettuce sitting next to condiments."

# Results: Clockwork for image editing

# The potential of generative video editing

Given an input video and a text prompt describing the edit, generate a new video

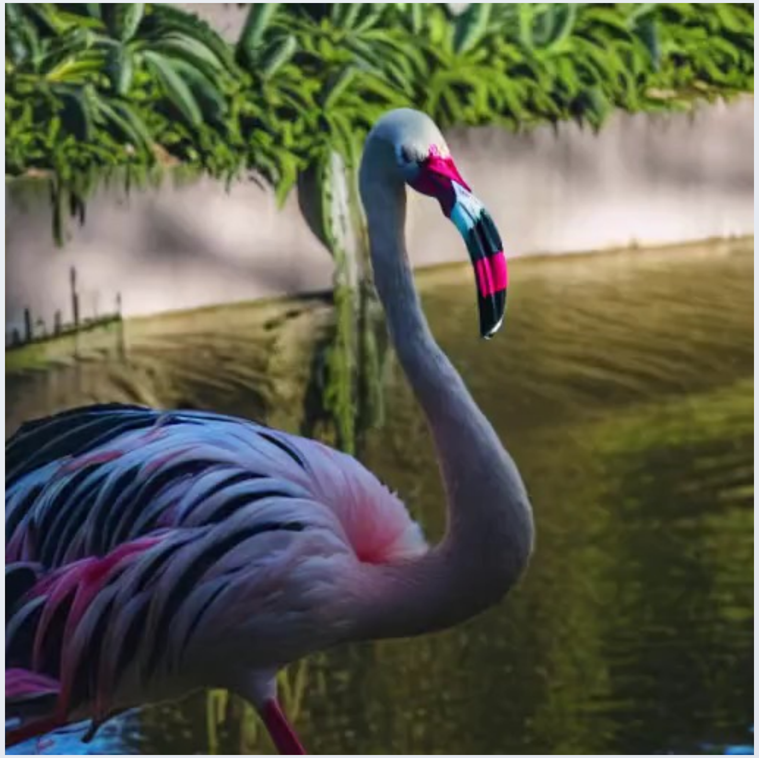
The edit usually changes the appearance or shape of a particular object

**Key challenges:**

1. Temporal consistency
2. High computational cost

Input video

Edited video

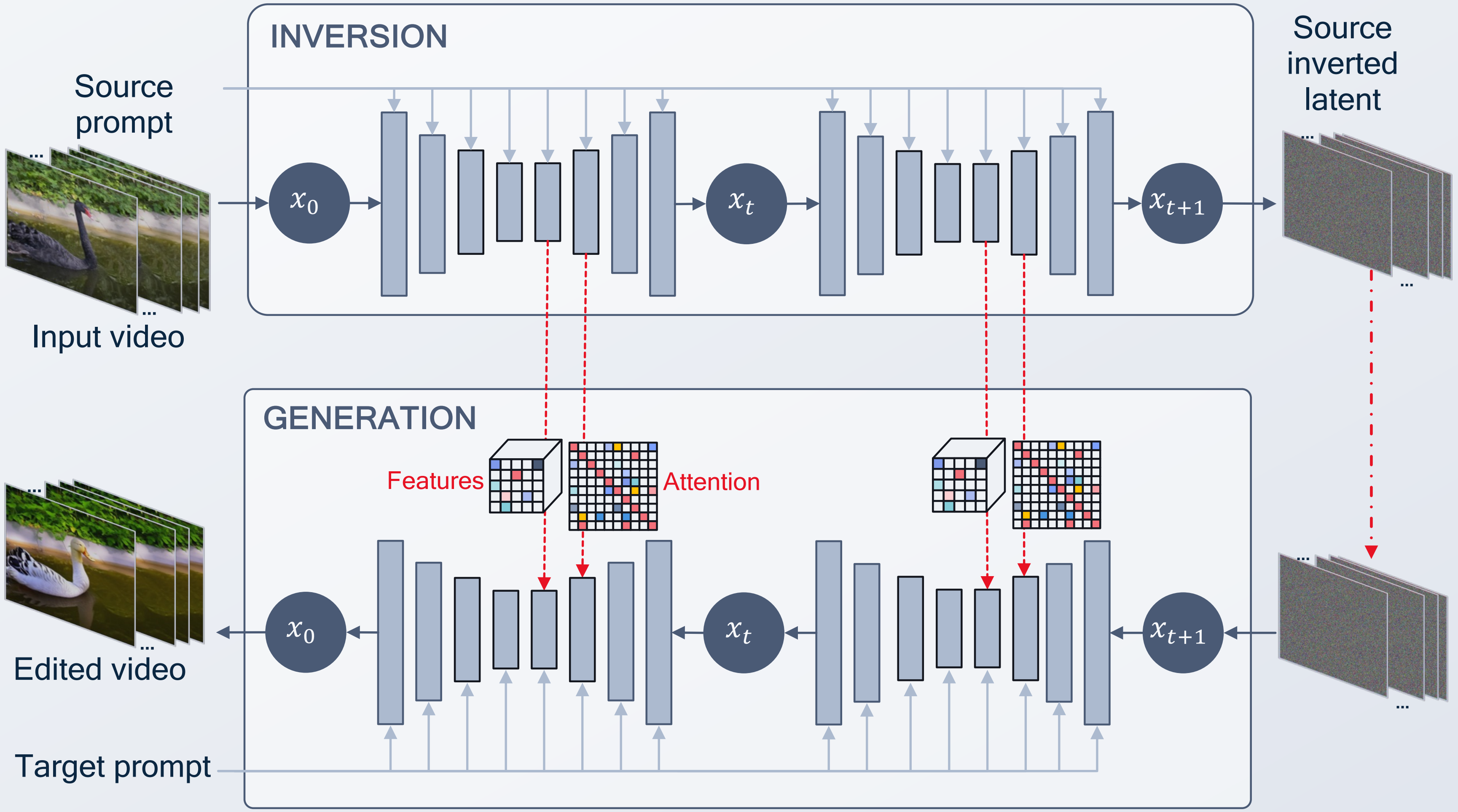


Prompt: "pink flamingo walking"

# Why is video editing so slow?

## Diffusion inversion

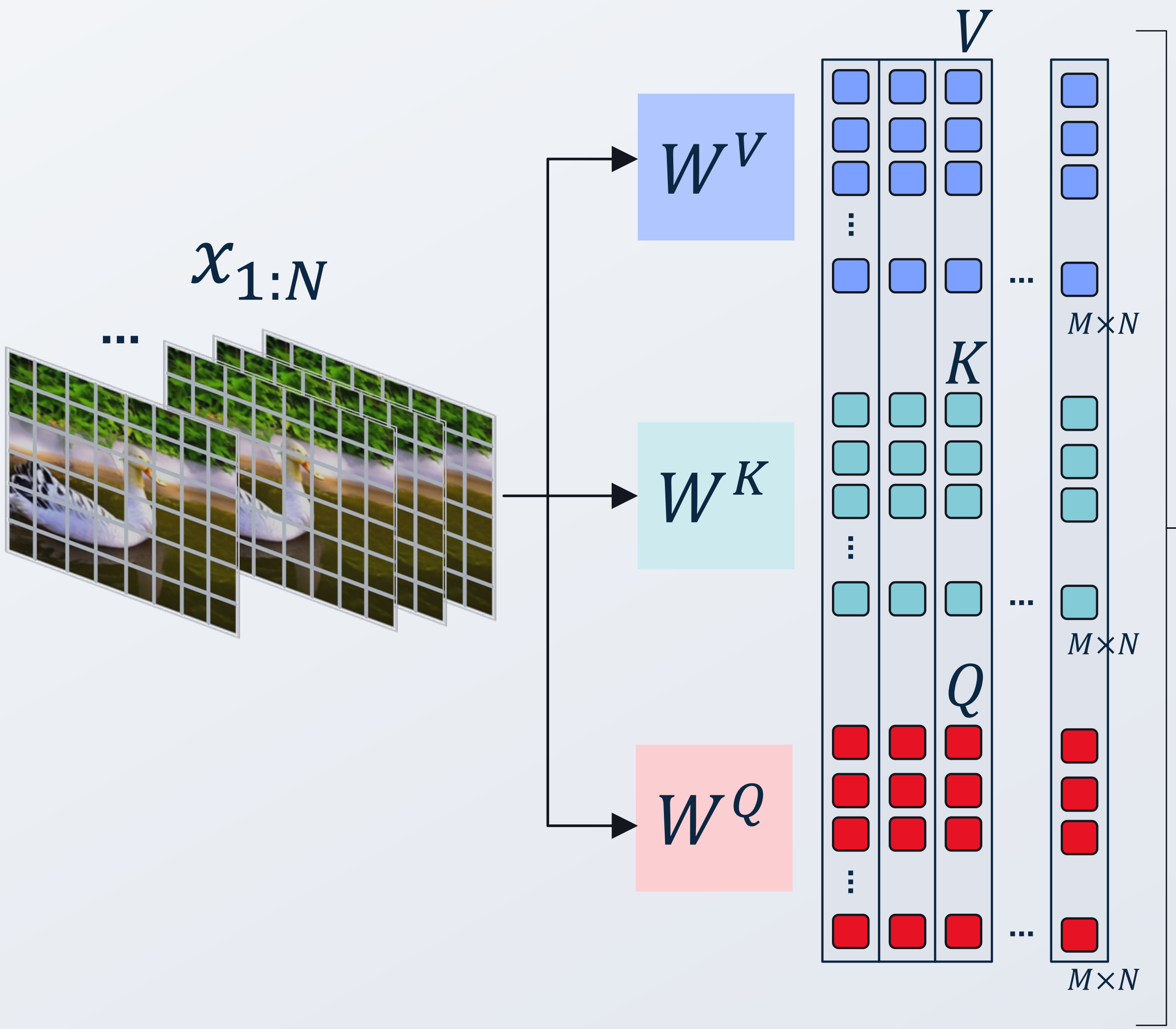
- Essential to preserve temporal consistency and details in the source video
- Comes at a high memory cost to store attention maps and feature



# Why is video editing so slow?

## Temporal attentions

Comes at a high computational cost due to the quadratic cost with respect to video length




$$\text{Softmax}\left(\frac{Q\,.\,K^T}{\sqrt{d}}\right).V$$

### Spatial Attention

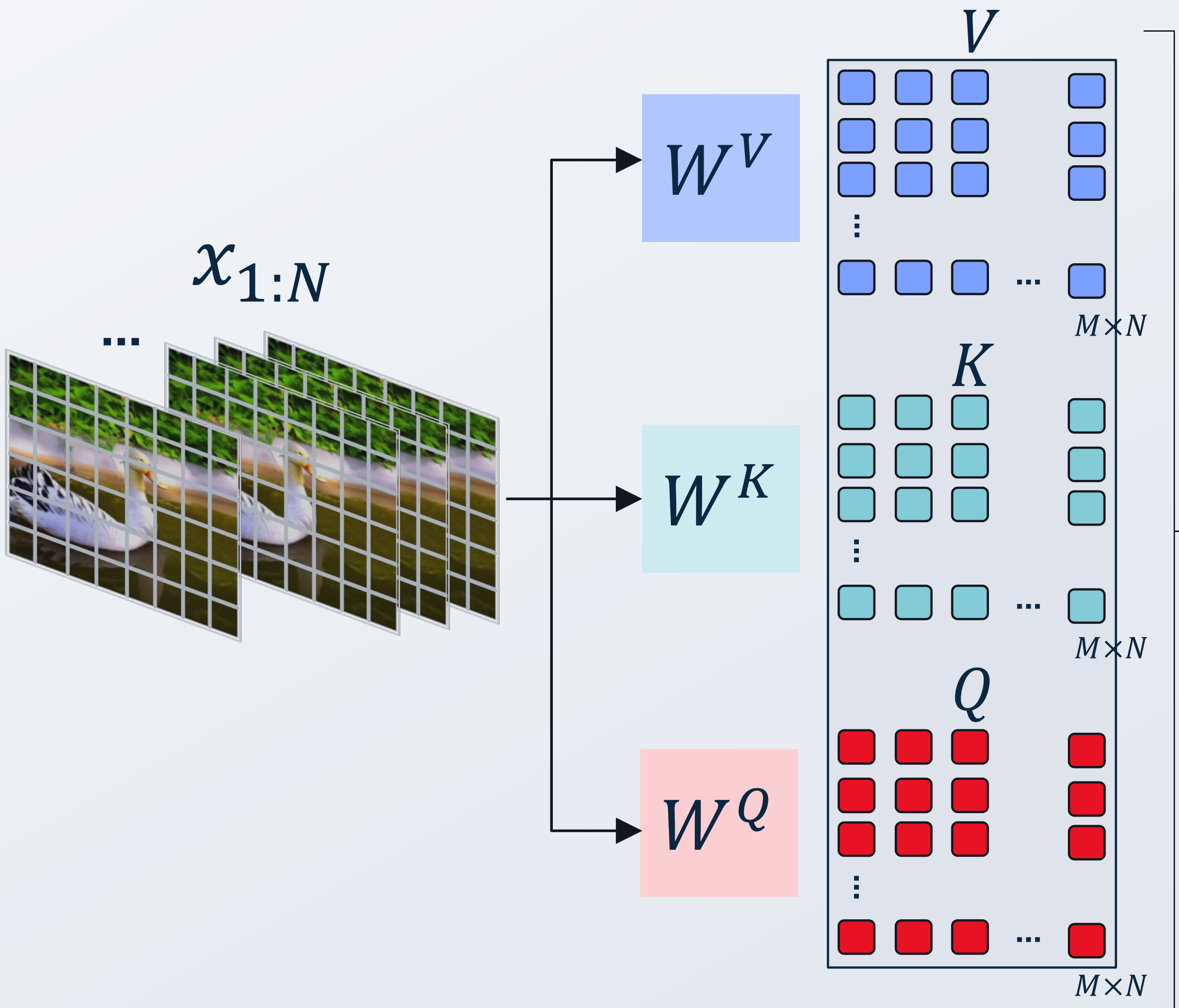
Compute self-attention individually per frame

Self-attention cost $O(M^2) \times N$

# Why is video editing so slow?

## Temporal attentions

Comes at a high computational cost due to the quadratic cost with respect to video length




Softmax $\left(\frac{Q \,.\, K^T}{\sqrt{d}}\right).V$

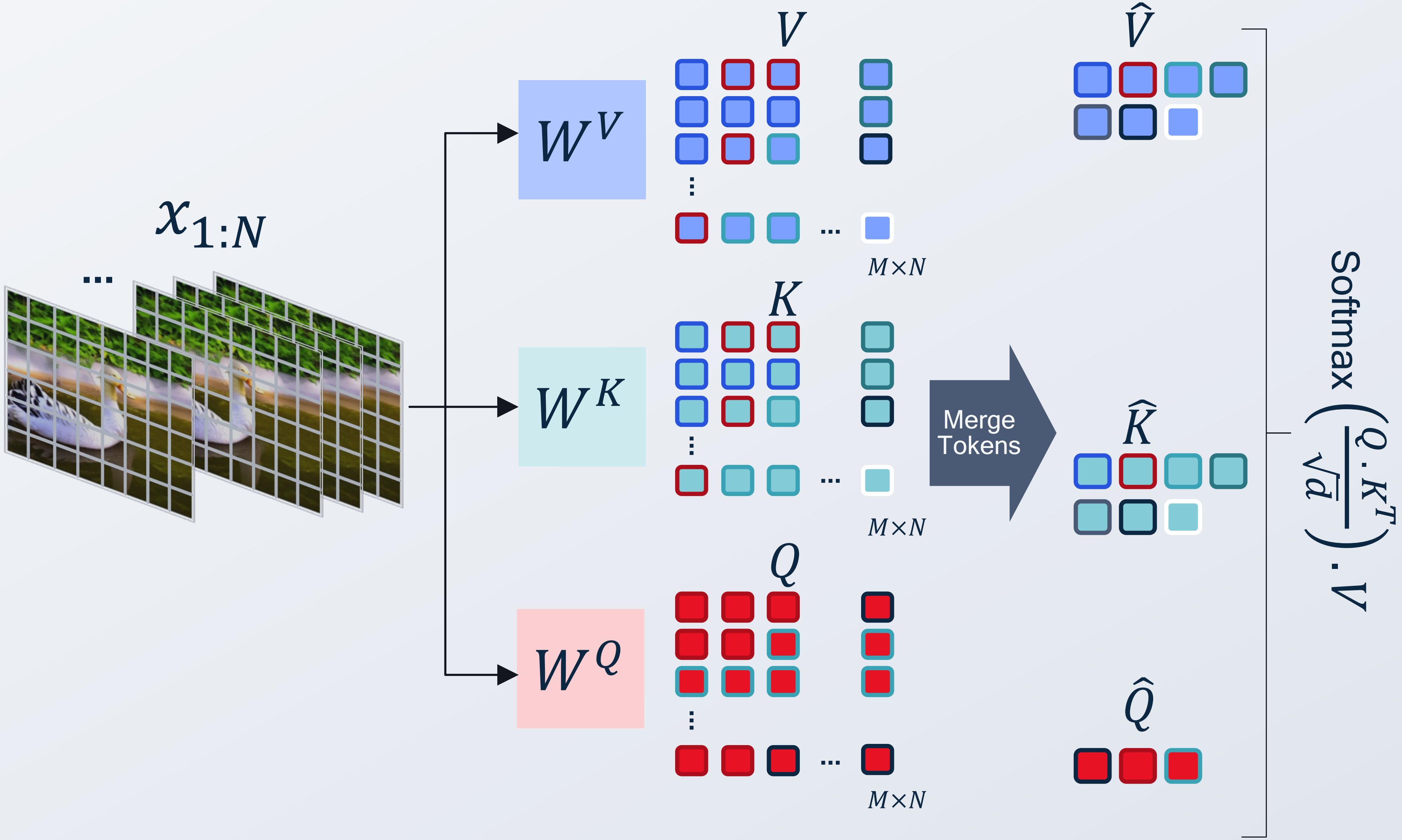
### Temporal Attention

Compute self-attention over all $N$ frames

Self-attention cost $O(M^2\ N^2)$

# Token merging solves these two challenges

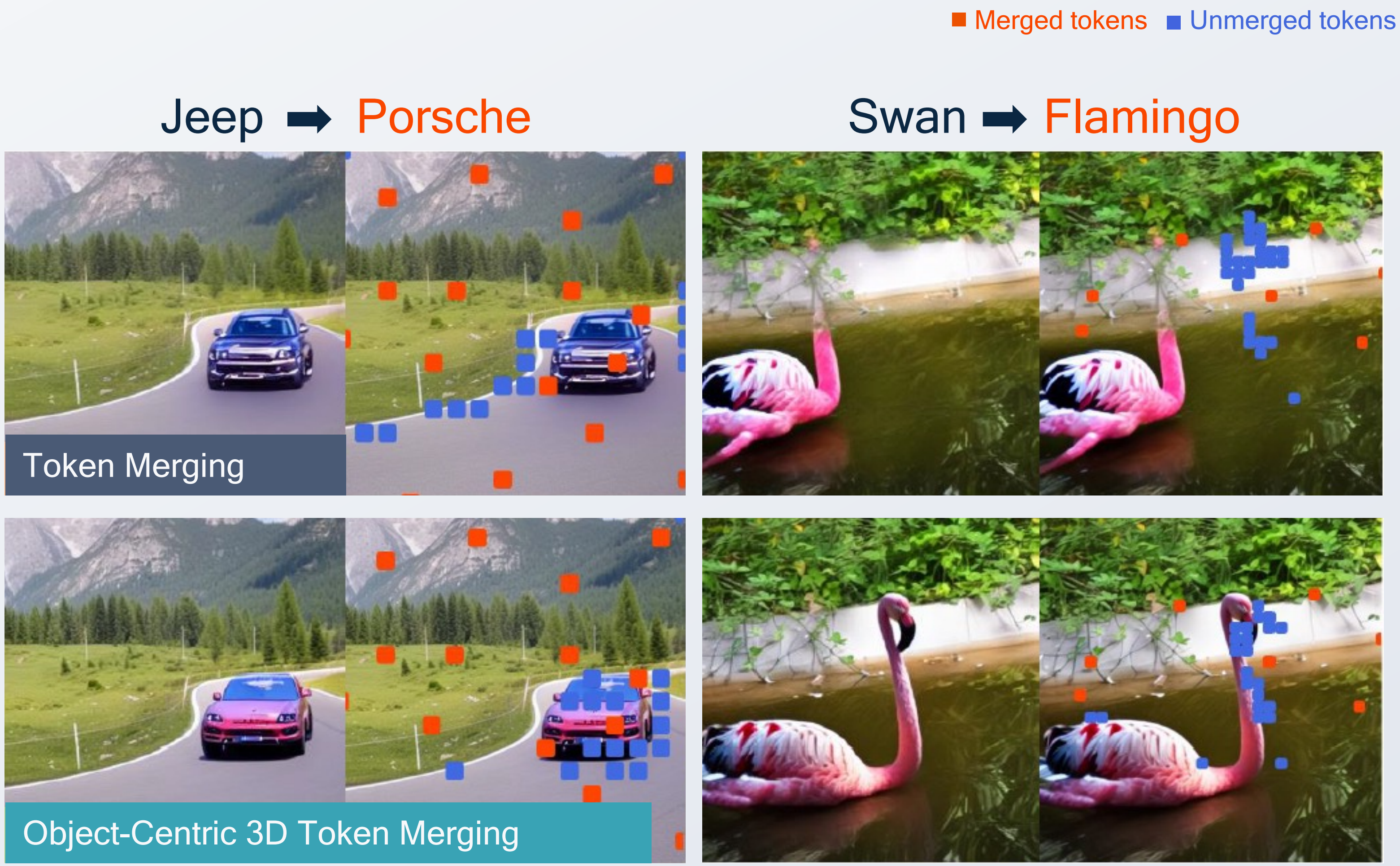
1. Merge the redundant tokens
2. Perform computation on clusters
3. Copy the output back into merged tokens




The costly attention is computed over a fraction of tokens (centroids)

# Encourages merging tokens on the background regions

By increasing $\eta$,
more and more
foreground tokens
will be left unmerged

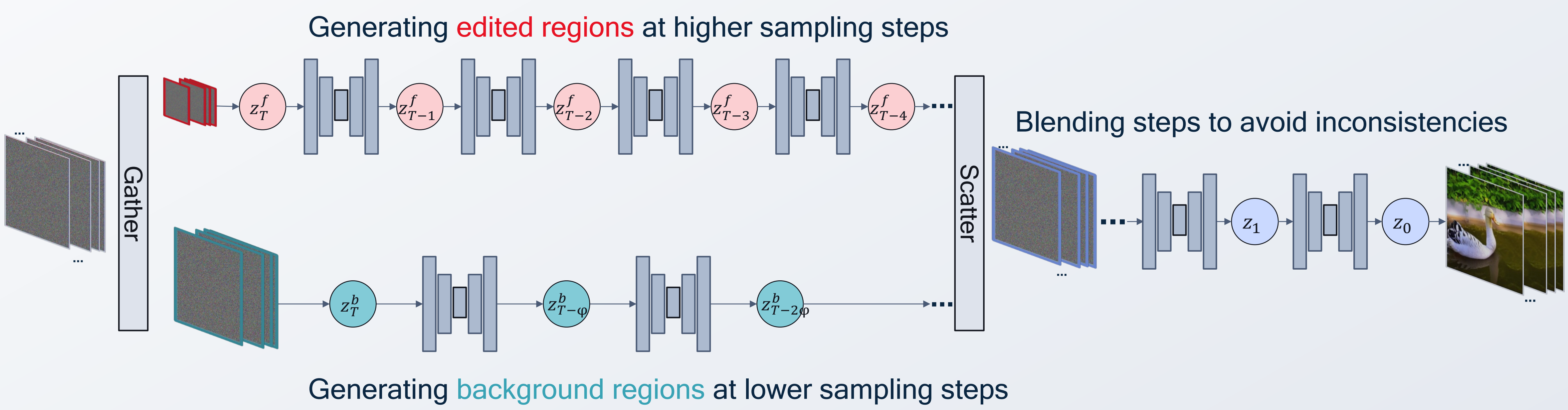



Two tokens are merged if their similarities exceeds a threshold

Introduce different thresholds for background vs. edited regions

Using lower threshold on background regions:

- Encourages merging more tokens on background
- Leaves more unmerged tokens on foreground regions




We perform a different number of sampling steps on edited and background regions:

Edited regions: Are usually small, but require most synthesis (more sampling steps)

Background regions: Are usually large, and don't require much synthesis (less sampling steps)

# Further acceleration by Object-Centric Sampling

# 6 -10x speedup with negligible drop in quality

**Applied our acceleration on two recent video generation frameworks:**

- FateZero
- ControlVideo

**Our acceleration includes:**

- Object-centric 3D token merging
- Object-centric sampling

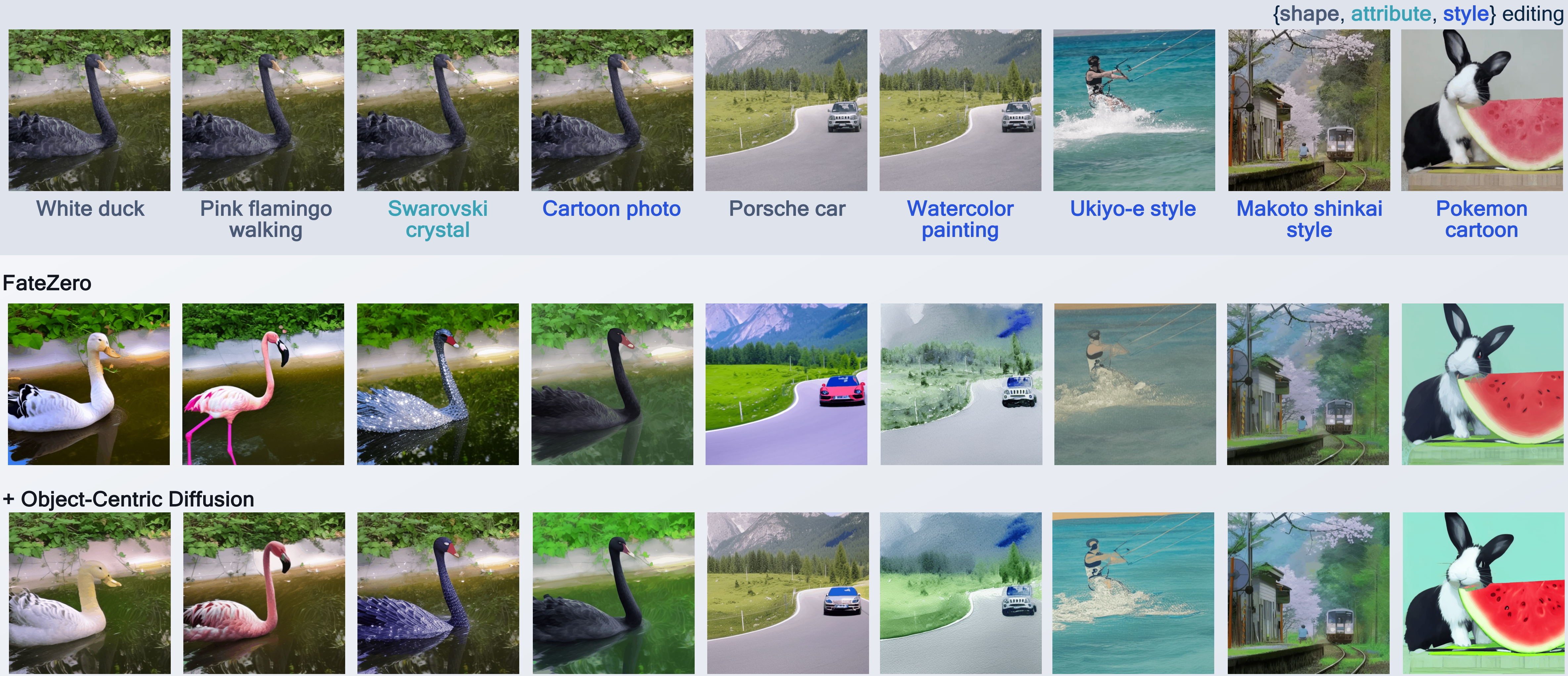
| Model | Temporal Cons ↑ | CLIP ↑ | Latency $(s)$ ↓ | | |
|---|---|---|---|---|---|
| | | | Inversion | Generation | Overall |
| FateZero | 0.961 | 0.344 | 135.80 | 41.34 | 177.14 |
| **+ Our acceleration** | 0.967 | 0.331 | 8.22 (16.5×) | 9.29 (4.4×) | 17.51 (10×) |

DAVIS benchmark: Generating 8 frames on server GPU

| Model | Temporal Cons ↑ | CLIP ↑ | Latency $(s)$ ↓ |
|---|---|---|---|
| ControlVideo | 0.972 | 0.318 | 152.64 |
| **+ Our acceleration** | 0.977 | 0.313 | 25.21 (6.0 ×) |

CV benchmark: Generating 15 frames on server GPU

FateZero: Fusing Attentions for Zero-shot Text-based Video Editing, ICCV'23
ControlVideo: Training-free Controllable Text-to-Video Generation, arXiv'23




# 10x faster at a comparable editing quality

Research on further optimizations to enable on-target deployment of video generation models

# How does 3D generation work?

Generating 3D mesh from a text prompt

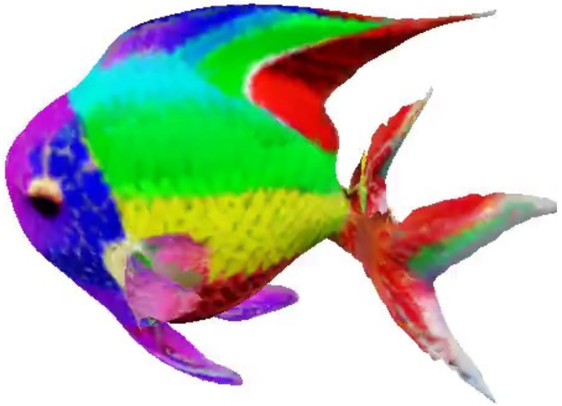
- Crucial for many tasks e.g., XR, graphics
- Manual creation of 3D assets is costly

A plush dragon toy

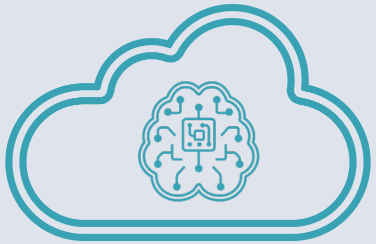
A DSLR photo of a hippo wearing a sweater

A DSLR photo of a train engine made out of clay

A beautiful rainbow fish



## Optimization-based approach



- Costly optimizations to fit mesh parameters for each object
- Takes **+20 min** to model a new object / scene
- Leverage a pretrained image generator to improve the optimization, i.e., score distillation sampling[1]

## Feed-forward approach

- Generate mesh parameters directly without any optimizations at inference
- Takes **seconds** to model a new object / scene
- Learned from scratch on the limited 3D data available

# Can pretrained image generators, e.g., Stable Diffusion, improve feed-forward 3D generation?

Transfer the huge diversity in 2D image datasets into 3D tasks

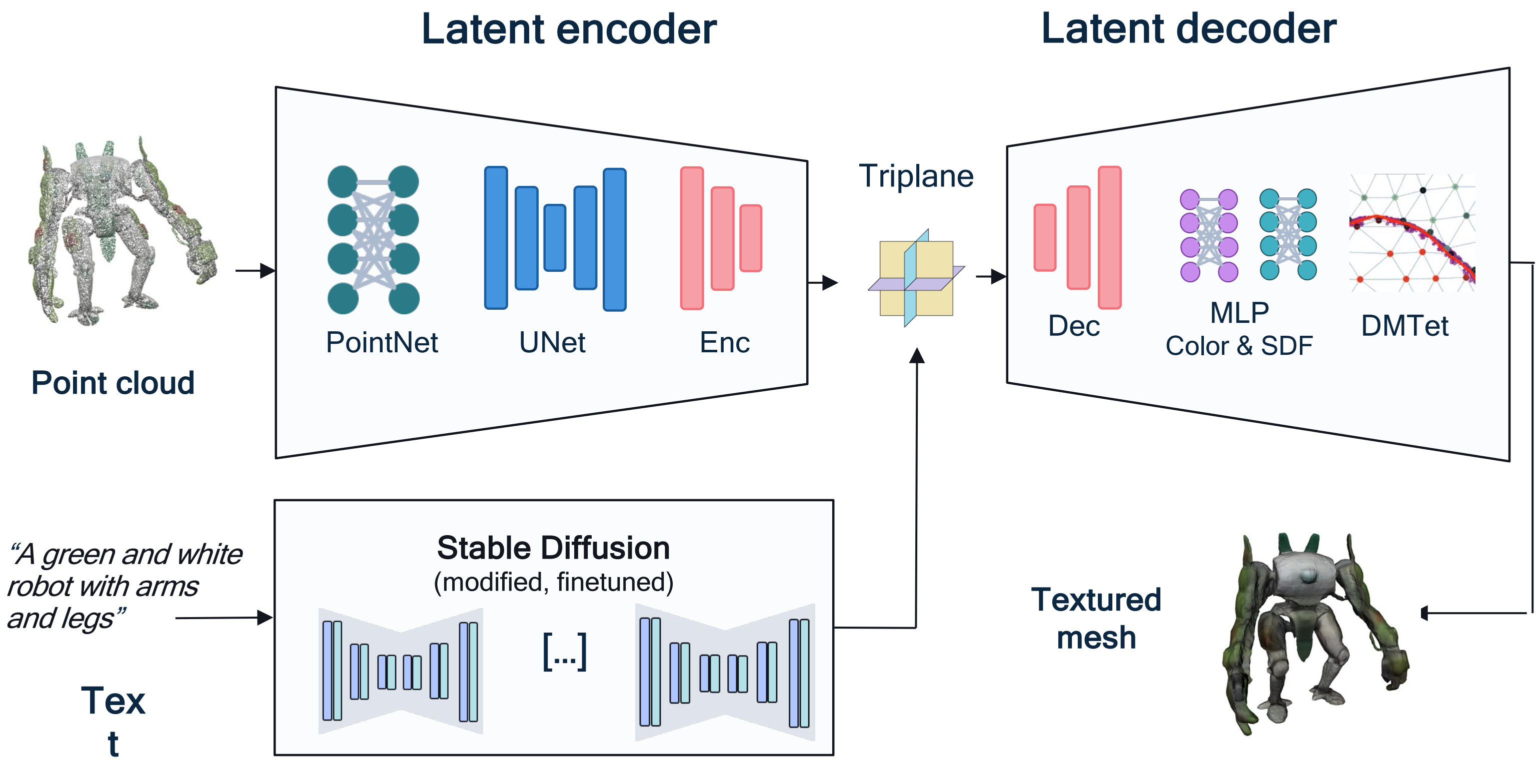
[1] DreamFusion: Text-to-3D using 2D Diffusion, arXiv'22

# HexaGen3D

Like other latent diffusion models, we follow a two-stage training:

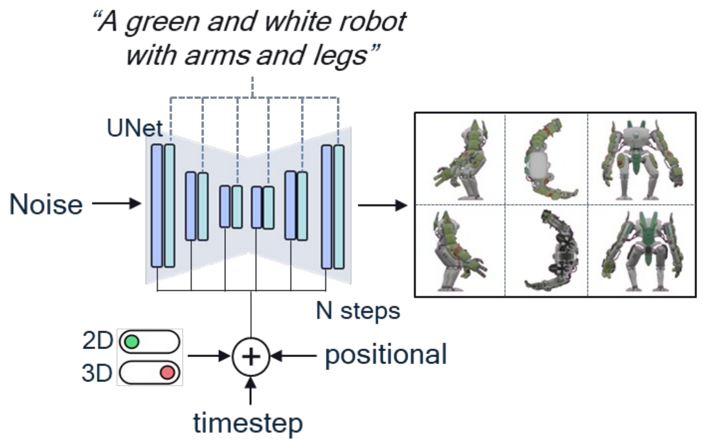
1. Variational Auto-Encoder (VAE) to reconstruct meshes from point clouds:
   - Latents defined in a triplane space of the shape $H \times 3.W \times C$

2. Conditional generation of triplanar latents from text
   - By adapting a pretrained Stable Diffusion model



## Step1:
## Generate Hexaview guidance

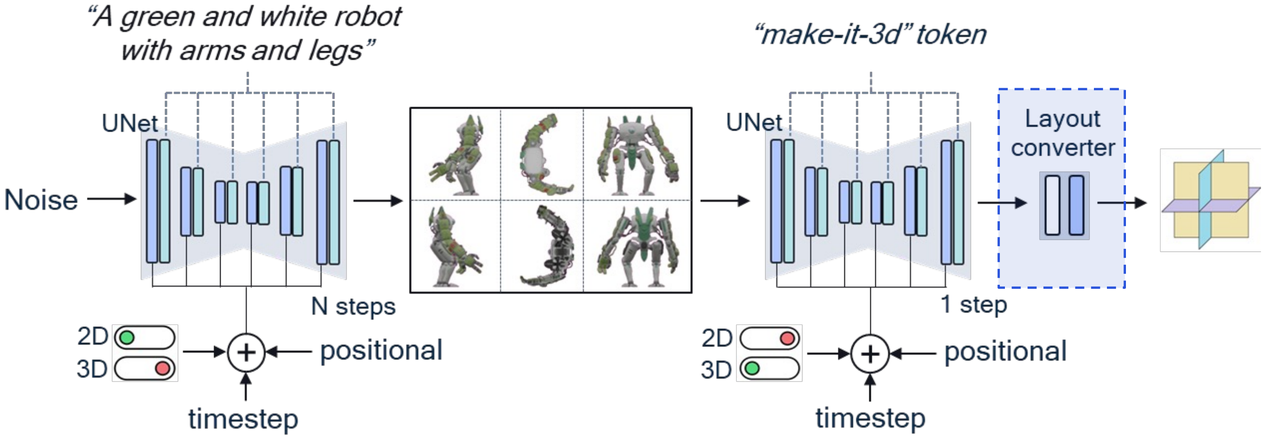
- Tile “front”, “rear”, “right”, “left”, “top” and “bottom” views into a large Hexaview image
- As an intermediate generation step, guides Stable Diffusion to generate triplanar latents




We generate triplanar latents using a pretrained Stable Diffusion Model in two steps

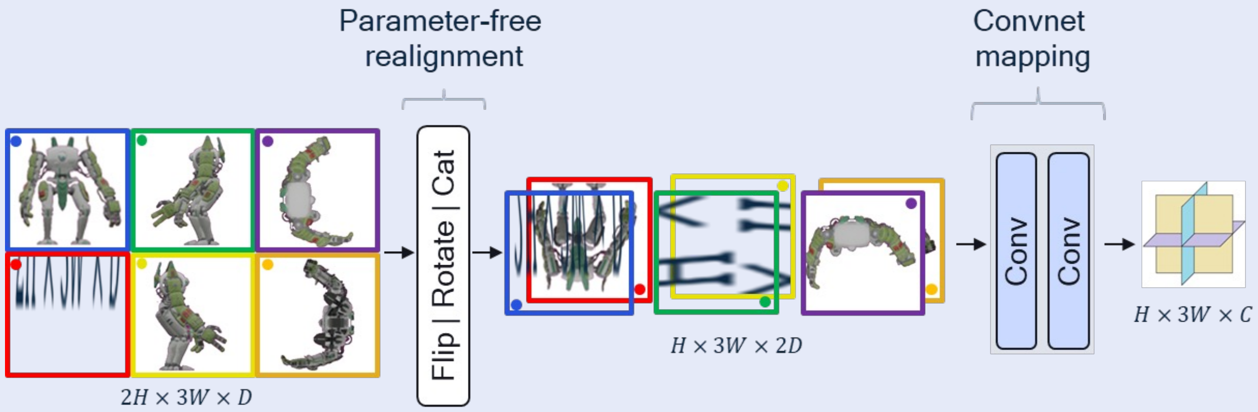
## Step1:
## Generate Hexaview guidance

- Tile “front”, “rear”, “right”, “left”, “top” and “bottom” views into a large Hexaview image
- As an intermediate generation step, guides Stable Diffusion to generate triplanar latents




## Step 2:
## Convert Hexaviews into triplanar latents

- Split and align the views, followed by a ConvNet
- Using the same UNet (parameter efficiency) with a different prompt and 3D embedding




We generate triplanar latents using a pretrained Stable Diffusion Model in two steps

# We generate high quality meshes, much faster than optimization-based methods:

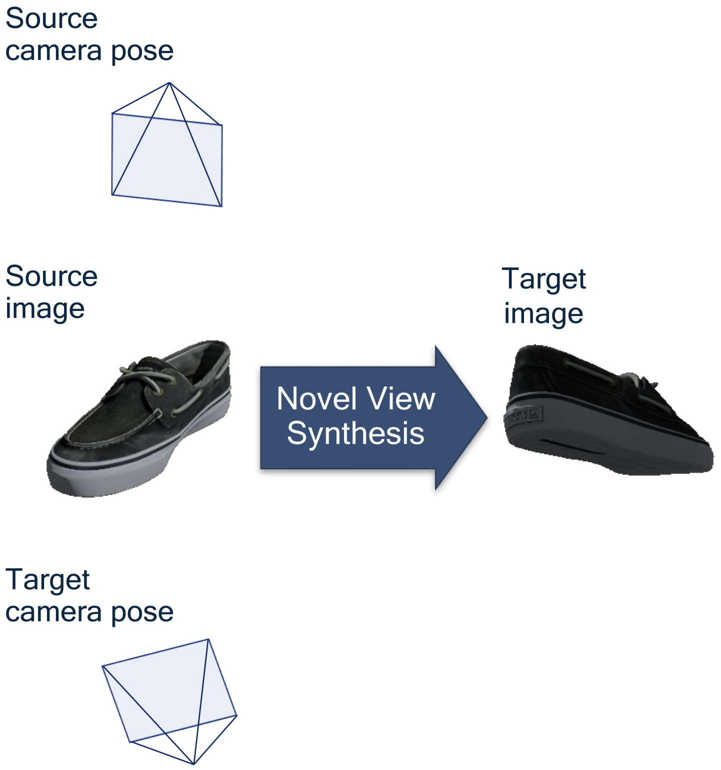
## 7 sec vs. +22 min

| Model | Latency ($s$) ↓ | CLIP ↑ | User preference↑ |
|---|---|---|---|
| MVDream | 194 mins | 30.35 | 0.97 |
| TextMesh | 23 mins | 25.06 | 0.12 |
| DreamFusion | 22 mins | 28.91 | 0.59 |
| HexaGen3D | 7 secs | 29.58 | 0.73 |


MVDream: Multi-view Diffusion for 3D Generation, arXiv'23
TextMesh: Generation of Realistic 3D Meshes From Text Prompts, arXiv'23
DreamFusion: Text-to-3D using 2D Diffusion, arXiv'22
Shape-E: Generating Conditional 3D Implicit Functions, arXiv'23

## Novel View Synthesis (NVS) generates a novel view of an object from a target pose

- Input: an **image** and a **camera pose**

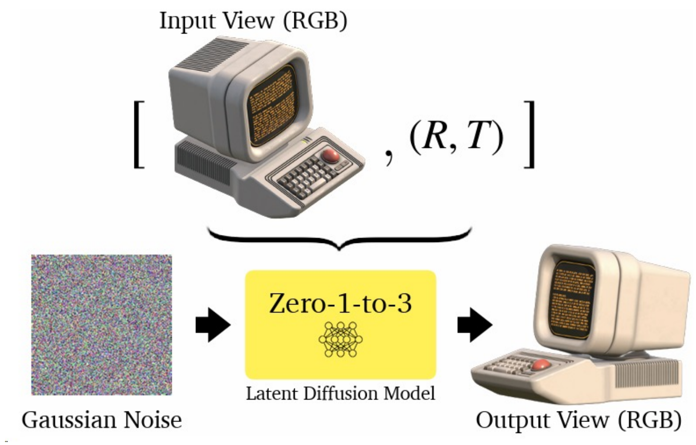



## Optimization based models i.e., NeRF

- Slow but high-quality when there are multiple views available

## Recently Stable Diffusion is adopted for Zero-shot NVS

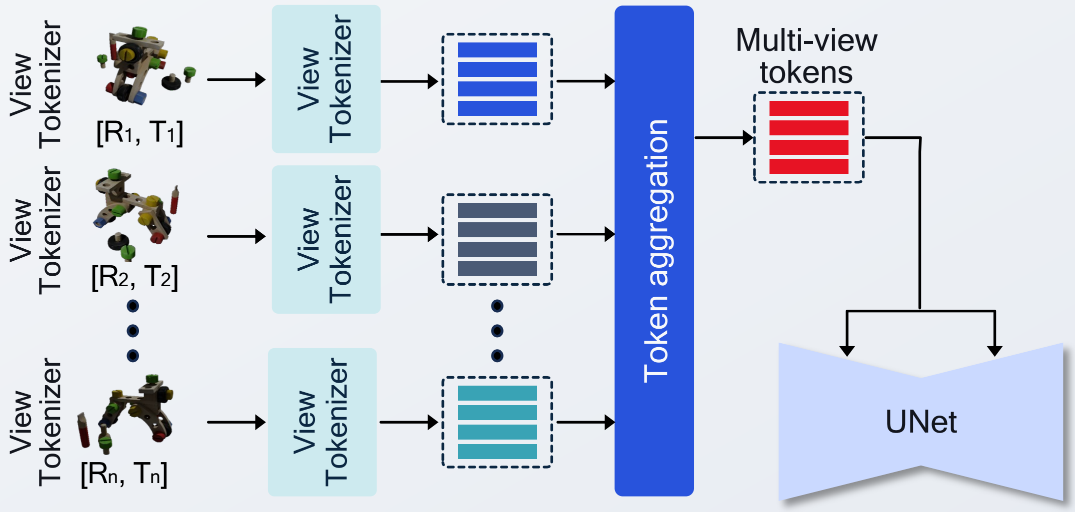
- Fast but quality is limited as consumes single view only




Zero-1-to-3: Zero-shot One Image to 3D Object, ICCV'23

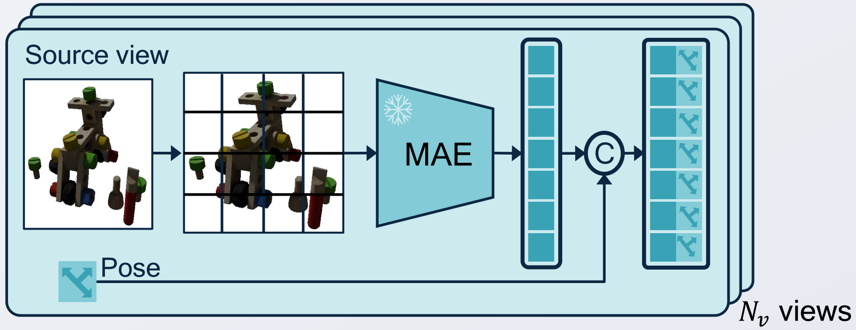
# How to enable zero-shot models to handle multiple views without increasing compute?

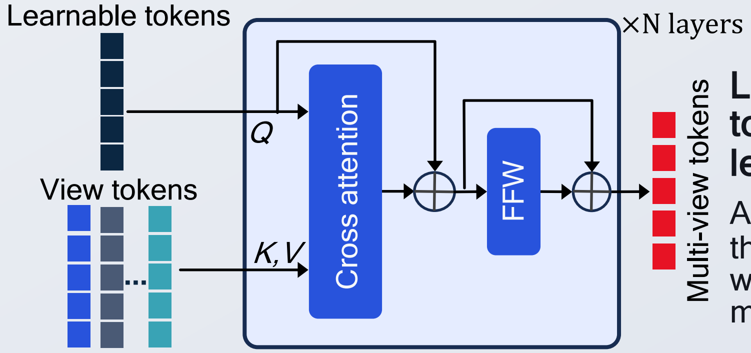
# Tokenize each view, aggregate over views, and use the multi-view tokens as cross-attention conditioning




**Pretrained Masked AutoEncoder (MAE) to tokenize views**

Provides better representation than CLIP




**Learnable tokens to generate fixed length output**

Avoid increasing the computations when fusing more views

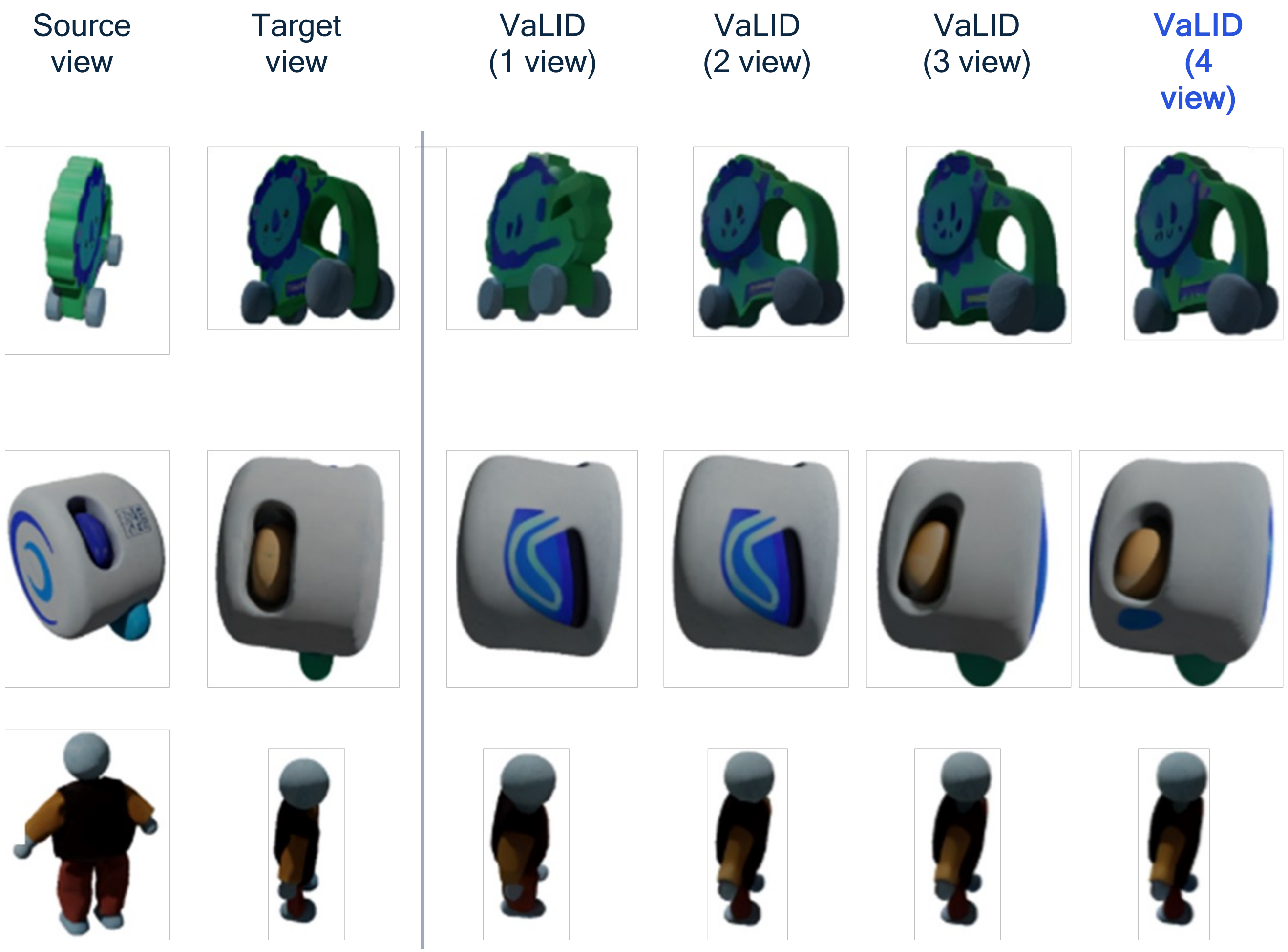
# VaLID: Variable Length Input Diffusion

# Our method outperforms existing SOTA methods in quality

## Google Scanned Object dataset

| Model | PSNR ↑ | LPIPS ↓ | GFLOPs ↓ |
|---|---|---|---|
| DietNeRF[1] | 8.93 | 0.412 | High |
| SJC-I[2] | 5.91 | 0.545 | High |
| IV[3] | 6.57 | 0.484 | High |
| Zero123[4] | 19.0 | 0.115 | Similar to ours |
| VaLID (1 view) | 20.03 | 0.091 | 87.2 |
| VaLID (2 view) | 20.41 | 0.085 | 87.8 |
| VaLID (3 view) | 21.05 | 0.073 | 88.8 |
| **VaLID (4 view)** | **21.30** | **0.069** | **91.4** |

PSNR = Peak Signal-to-Noise Ratio
LPIPS = Learned Perceptual Image Patch Similarity




At a negligible computational cost, VaLID processes multiple views to generate more accurate views

1: Putting NeRF on a Diet: Semantically Consistent Few-Shot View Synthesis, CVPR'21
2: Stable Diffusion Image Variation, arXiv'23
3: Score Jacobian Chaining: Lifting pretrained 2d diffusion models for 3d generation, CVPR'23
4: Zero-1-to-3: Zero-shot One Image to 3D Object, ICCV'23

# Our method outperforms existing SOTA methods in quality

## Google Scanned Object dataset

| Model | PSNR ↑ | LPIPS ↓ | GFLOPs ↓ |
|---|---|---|---|
| DietNeRF[1] | 8.93 | 0.412 | High |
| SJC-I[2] | 5.91 | 0.545 | High |
| IV[3] | 6.57 | 0.484 | High |
| Zero123[4] | 19.0 | 0.115 | Similar to ours |
| VaLID (1 view) | 20.03 | 0.091 | 87.2 |
| VaLID (2 view) | 20.41 | 0.085 | 87.8 |
| VaLID (3 view) | 21.05 | 0.073 | 88.8 |
| **VaLID (4 view)** | **21.30** | **0.069** | **91.4** |


At a negligible computational cost, VaLID processes multiple views to generate more accurate views

PSNR = Peak Signal-to-Noise Ratio
LPIPS = Learned Perceptual Image Patch Similarity

1: Putting NeRF on a Diet: Semantically Consistent Few-Shot View Synthesis, CVPR'21
2: Score Jacobian Chaining: Lifting pretrained 2d diffusion models for 3d generation, CVPR'23
3: Stable Diffusion Image Variation, arXiv'23
4: Zero-1-to-3: Zero-shot One Image to 3D Object, ICCV'23

# We adapt the generative model to new domain: automotive

Generative models improve graphic simulators by being:

- Realistic by being trained on real images and videos
- Scalable by sampling examples instead of manually crafting the assets/objects and scenario

### Generate training data for long-tailed object classes

i.e., animals and emergency vehicles

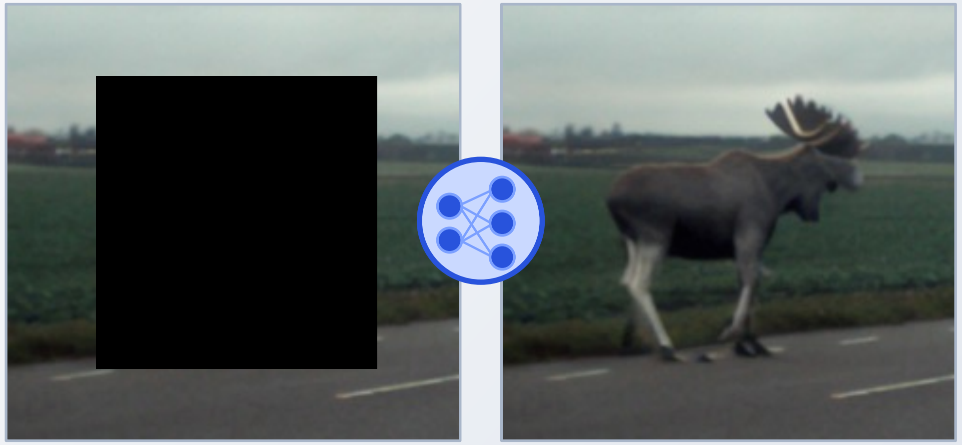
### Scale up test set by diversifying

i.e., weather, object appearance and geometries

### Generate safety critical test scenarios

i.e., crashes and pedestrians on road

## Animal Detection



| Training data | mAP |
|---|---|
| Real images | 50.2 |
| Real + generated images | 57.7 |

Qualcomm

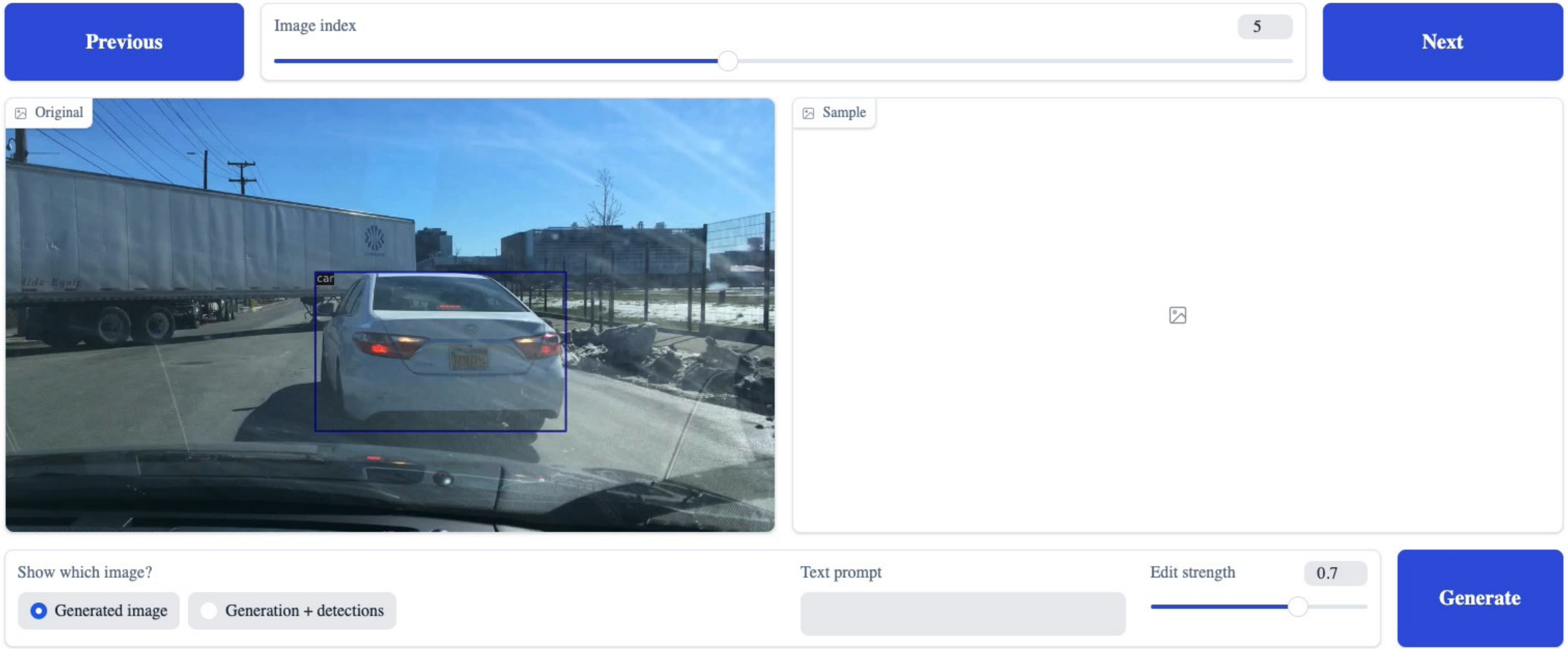

## Inpainting for animal detection

Adapting the generative model to new domains, i.e., automotive scenes

High-fidelity generation in tight bounding boxes

Putting animals at the right geometry: location and scale

Added to the training set to improve animal detector




## High-fidelity object editing

Using generative editing models to change appearance of vehicles

Diversify the test data to less common vehicle types like classics

Avoid unintended changes in appearance and geometry of vehicle and its background

Qualcomm

Generative vision has a great potential in image and video generation across enterprise, entertainment, XR, and automotive.

Efficient generative vision is important for achieving scale, at the cloud and on device.

Qualcomm AI Research has achieved state-of-the-art results in image and video generation with novel techniques.

Questions

# Connect with us

www.qualcomm.com/research/artificial-intelligence

www.qualcomm.com/news/onq

www.youtube.com/c/QualcommResearch

@QCOMResearch

https://assets.qualcomm.com/mobile- computing-newsletter-sign-up.html

www.slideshare.net/qualcommwirelessevolution

# Thank you

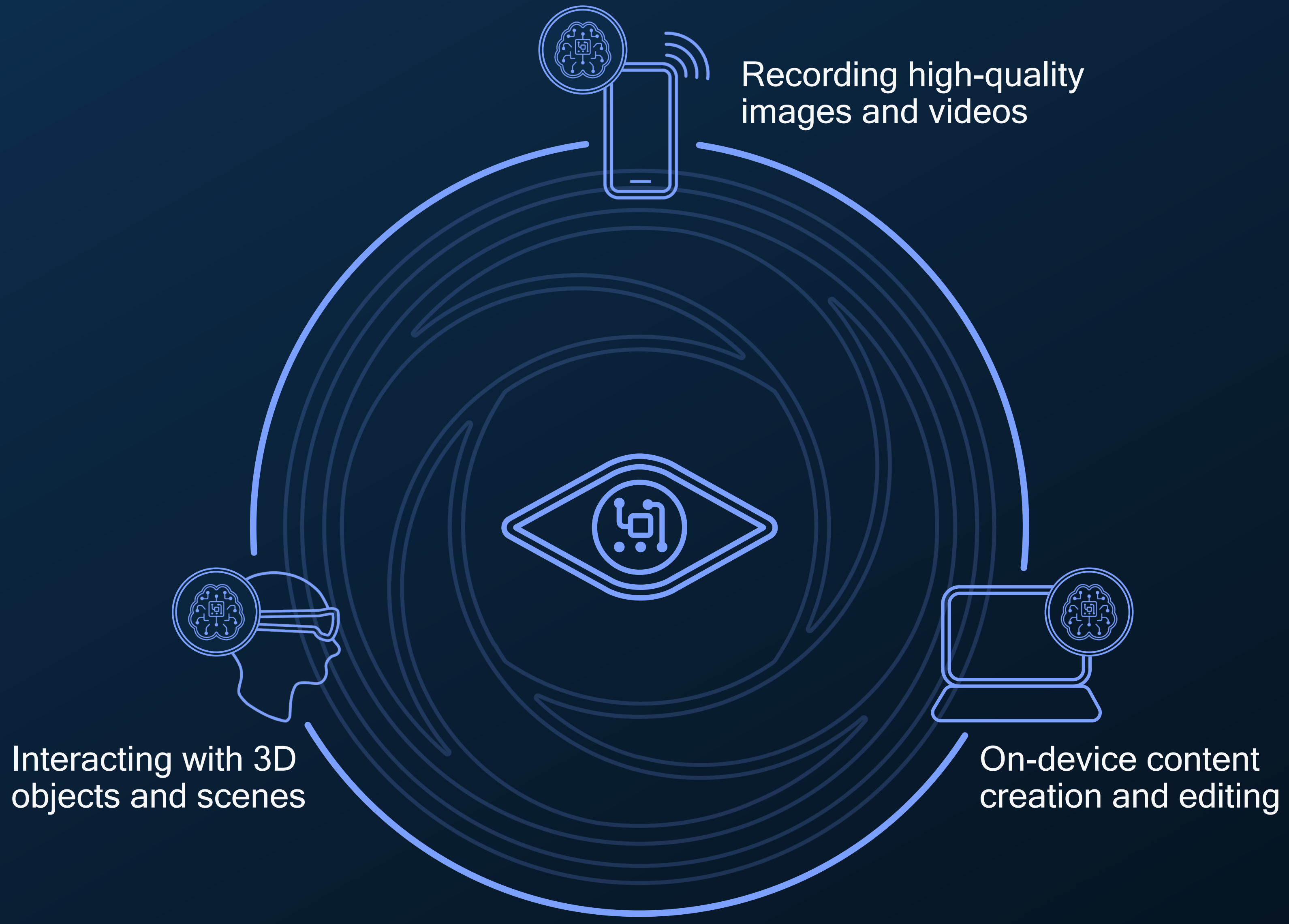
Follow us on:

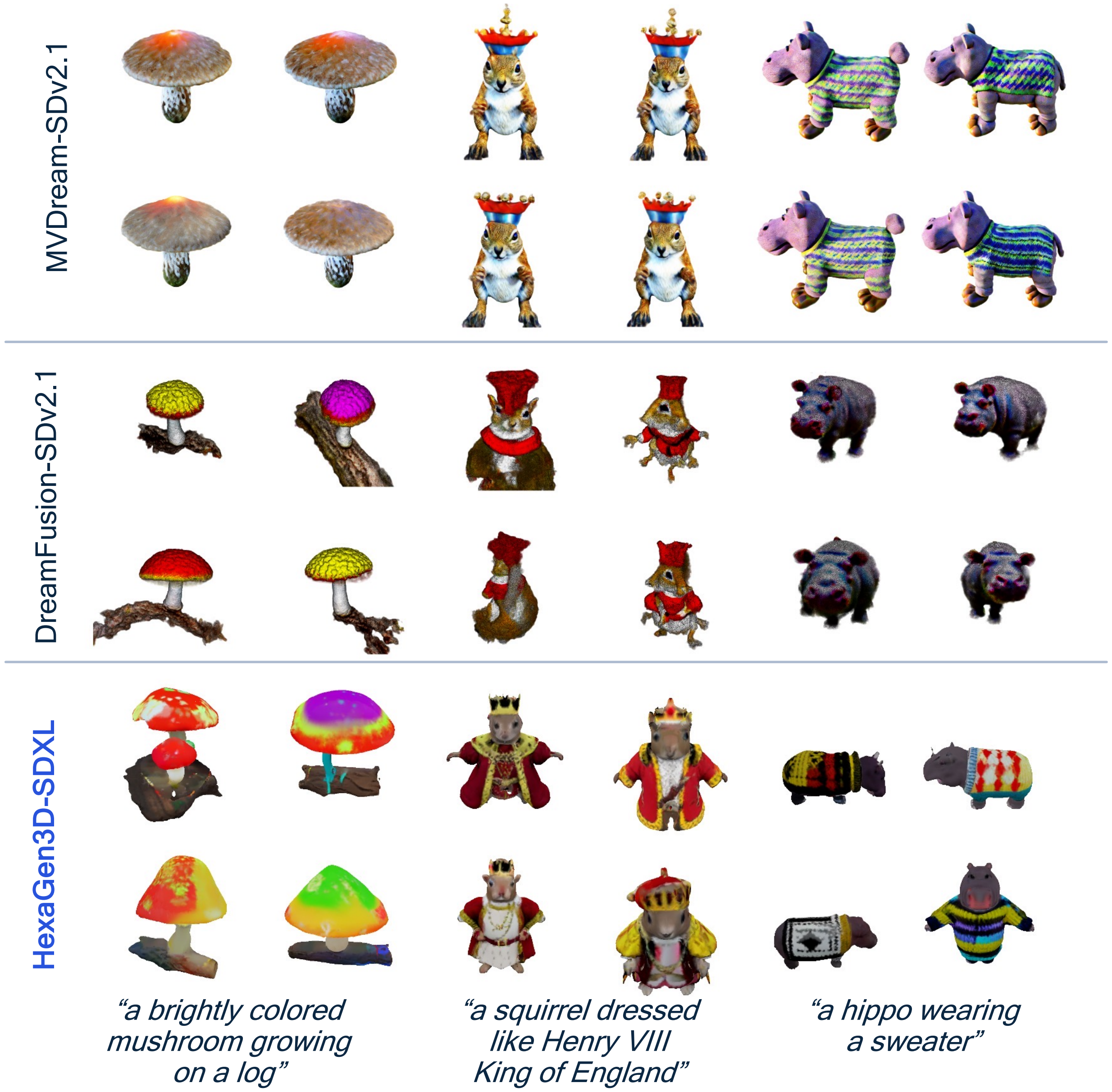
For more information, visit us at:

qualcomm.com & qualcomm.com/blog

# Efficient generative vision

Processing on the edge enables scale across devices






We generate more diverse generations (random seeds)

# Higher quality 3D images with HexaGen3D

Generating Hexaviews is much more effective than directly generating the triplanar latents

| CLIP score ↑ | 24.02 With Hexaview generation | 18.47 Without Hexaview generation |
|---|---|---|

Using the same UNet for generating and converting Hexaviews is more effective

| CLIP score ↑ | 24.02 With weight sharing | 23.43 Without weight sharing |
|---|---|---|