# Text-to-3D Generative AI on Mobile Devices: Measurements and Optimizations

Xuechen Zhang[1]*, **Zheng Li**[2]*, Samet Oymak[2], Jiasi Chen[2]

[1]University of California, Riverside
[2]University of Michigan, Ann Arbor
*co-first authors

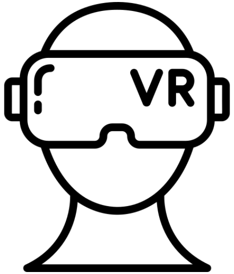
# Text-to-3D Generative AI

"A pumpkin" ⟹ **Text-to-3D Generative AI** ⟹

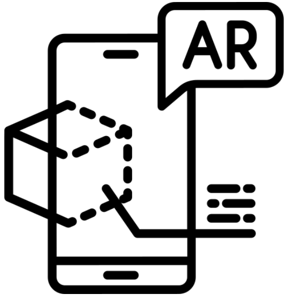
**Input**: Text Prompt

**Output**: 3D Model

## Application Scenarios:

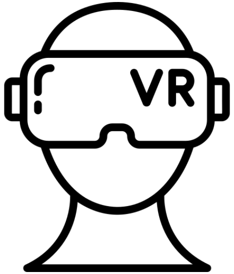



**Gaming**

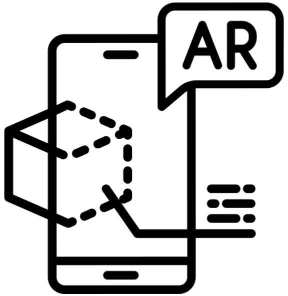



**Product Design**

# Text-to-3D Generative AI




Gaming




Product Design

→ Mobile Devices

**Problems:**

**Not ready for mobile deployment due to resource constraints (memory, compute, energy, etc.)**

**E.g., DreamFusion takes 12 hours to generate a 3D object on a NVIDIA V100 GPU**

# Motivation

**We want to deploy Text-to-3D generative AI on mobile devices while ensuring good user experience**

→ **Low Latency**
**Low Memory Usage**
**High 3D Object Synthesis Quality**

# Motivation

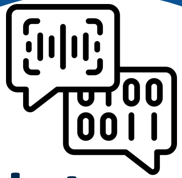
Low Latency
Low Memory Usage
High Synthesis Quality

Optimization

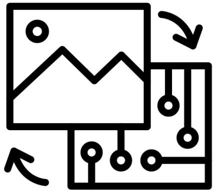
Measurements to identify bottlenecks

# Background

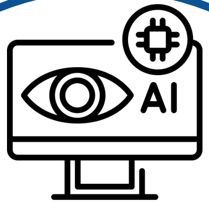
## Text - to - 3D



**Natural Language Processing**



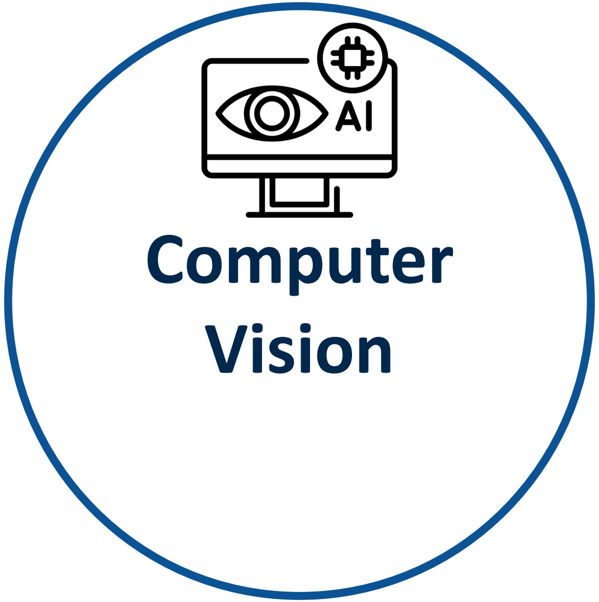
**Generative AI**




**Computer Vision**

# Background: 3D Representations

3D

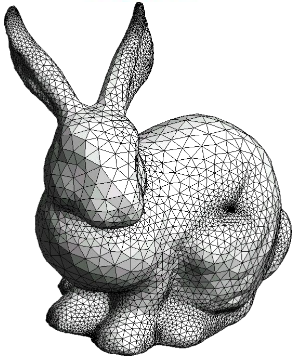


# Background: 3D Representations

## Explicit Representation

**Point Clouds**

**3D Meshes**



## Implicit Representation

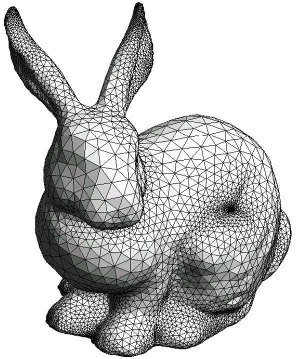
**NeRF**

**SDF**

# Background: Explicit Representations

**Point Clouds**

**3D Meshes**

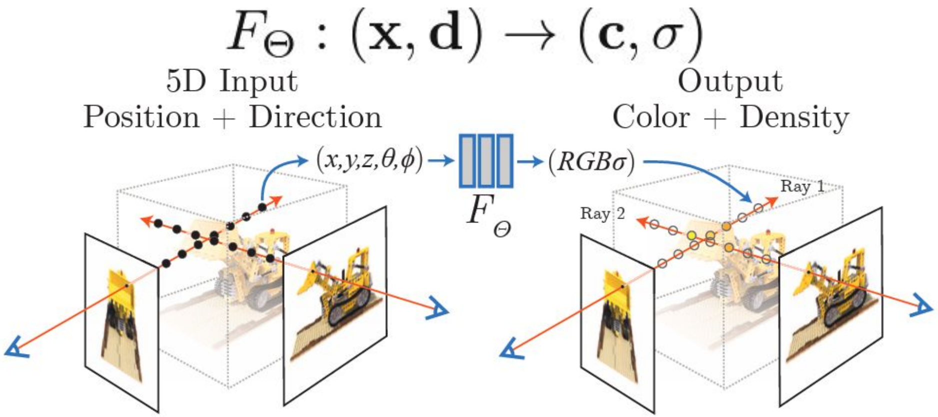


**Usually use discrete locations represented by points, edges etc.**

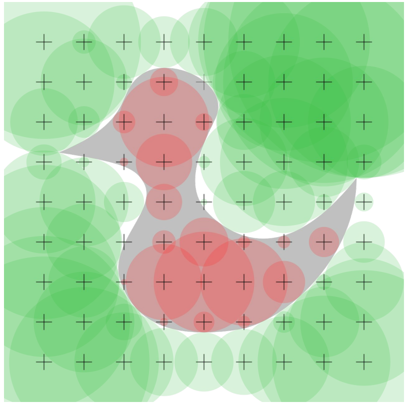
Low Latency
Low Memory Usage
Low Synthesis Quality

# Background: Implicit Representations

## NeRF: Neural Radiance Fields




## SDF: Signed Distance Field



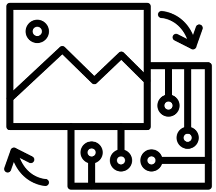
**High Latency due to Computation**
**High Memory Usage due to Computation**
**High Synthesis Quality**

# Background: 3D Representations

| | Explicit Representations | Implicit Representations |
|---|---|---|
| Latency | ↓ | ↑ |
| Memory Usage | ↓ | ↑ |
| Synthesis Quality | ↓ | ↑ |

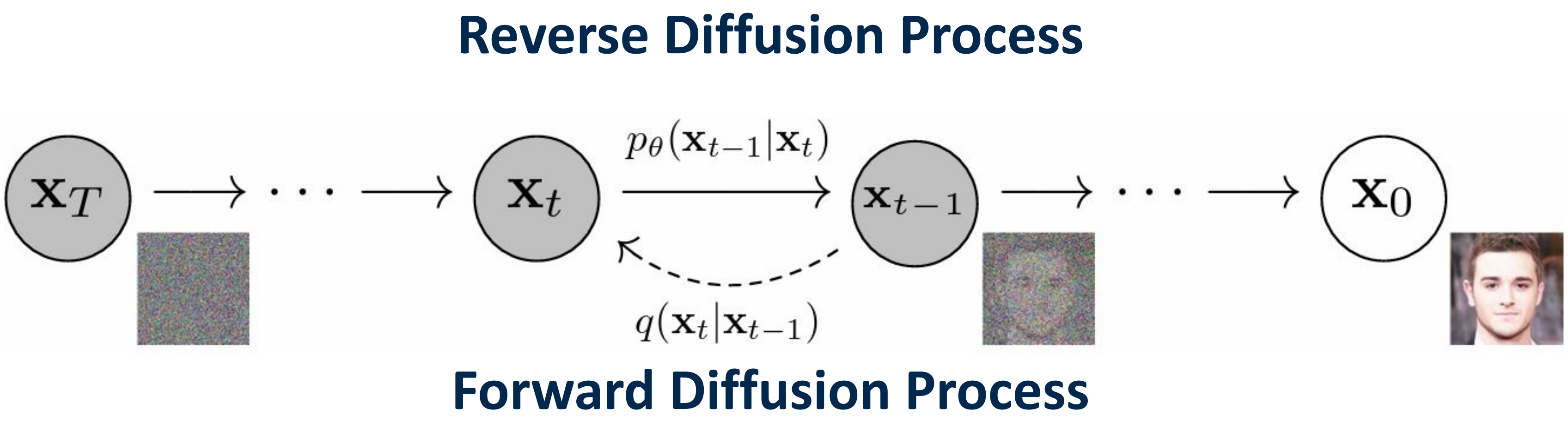
# Background

to



Generative AI

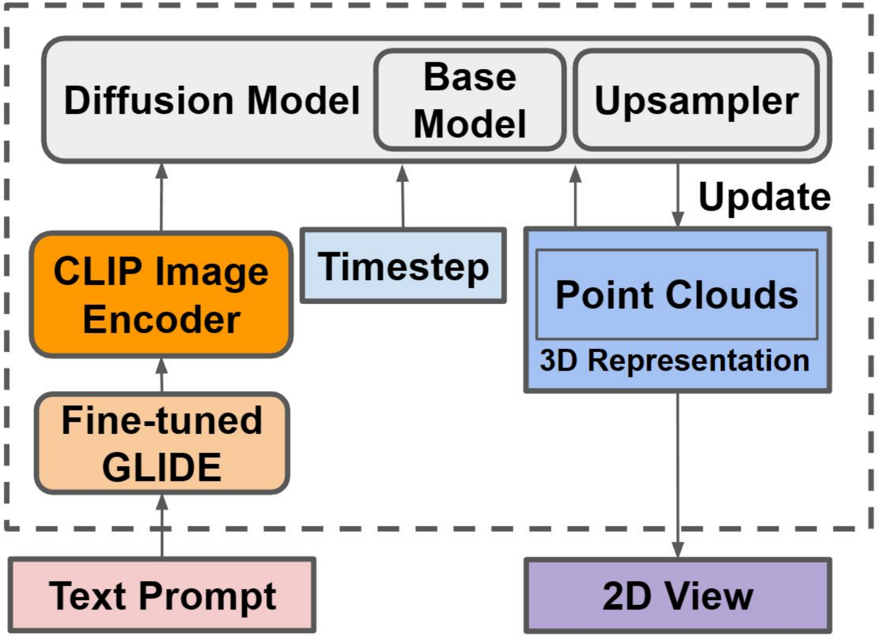
# Background: Diffusion Model



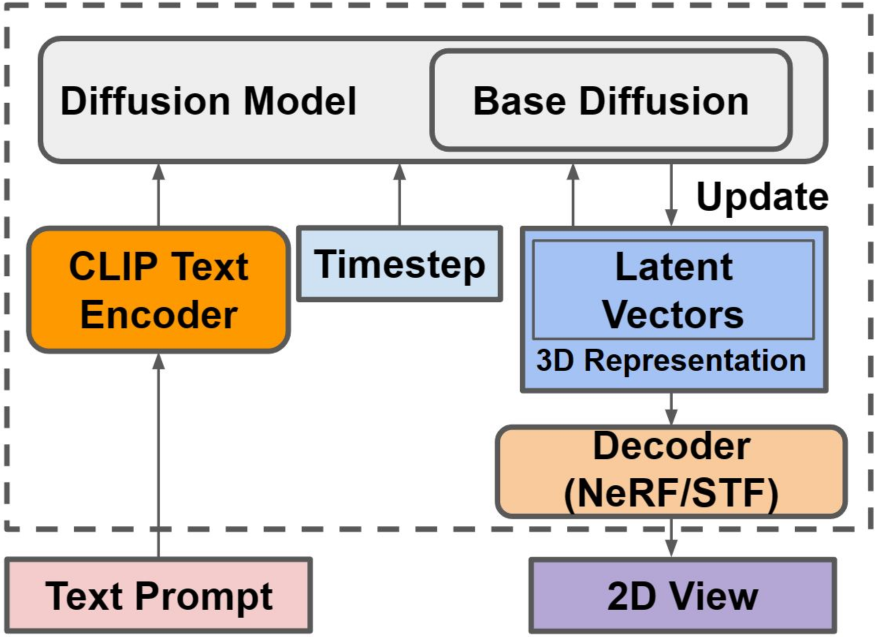

**Many steps** of an **expensive** machine learning model (e.g. Unet, ViT) is needed to learn the reverse diffusion process.

# Diffusion Model Overview

## Point-E (Dec. 2022)




## Shap-E (May 2023)



# Measurements

What are the **bottlenecks** to deploy text-to-3D models on mobile devices?

What to measure?

Optimization Goals:
Low Latency
Low Memory Usage
Good Synthesis Quality

# Measurement Setup

## Hardware:

**NVIDIA T4 GPU (weak server GPU)**
**NVIDIA Jetson AGX Orin (mobile GPU)**

## Dataset:

COCO
Common Objects in Context

# Measurement Setup: Model Configurations

## For Point-E and Shap-E:

### Parameter count for Diffusion:

- 40M
- 300M
- 1B

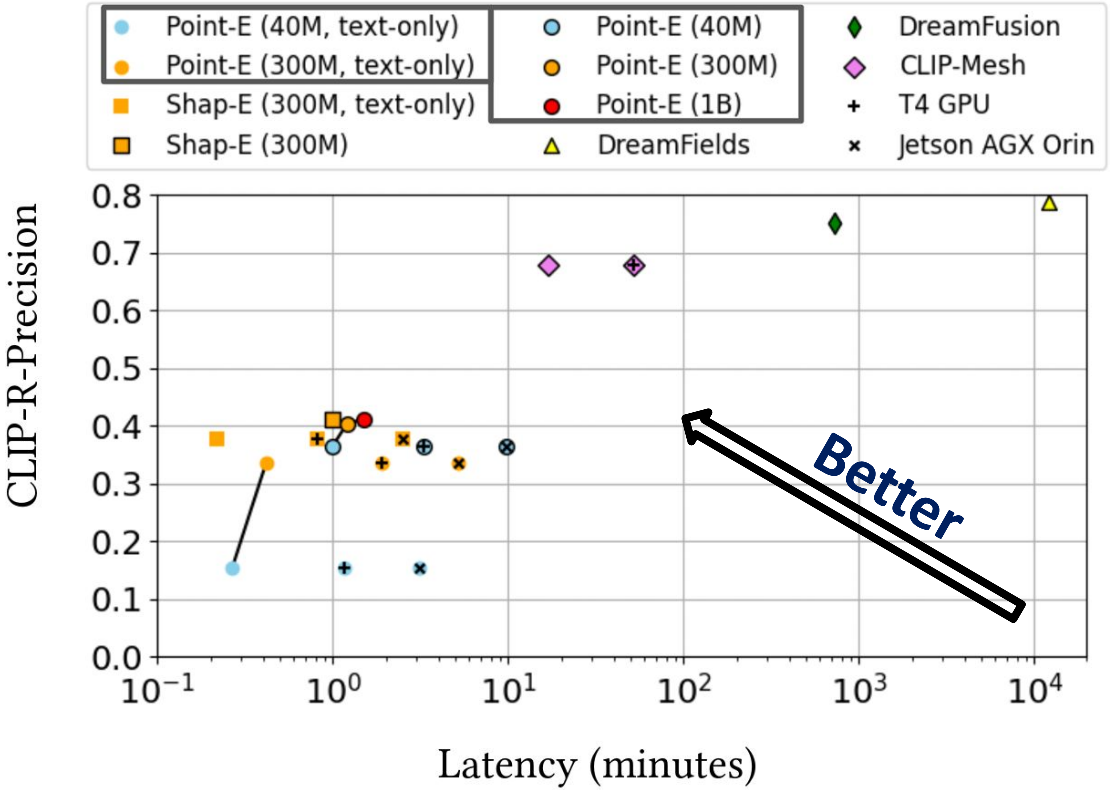
### Conditioning options:

- Text-only
  Text → 3D
- Image-conditional (Default)
  Text → 2D → 3D

# Latency-Quality Tradeoff

**Synthesis quality:**
**Image-conditional > Text-only**

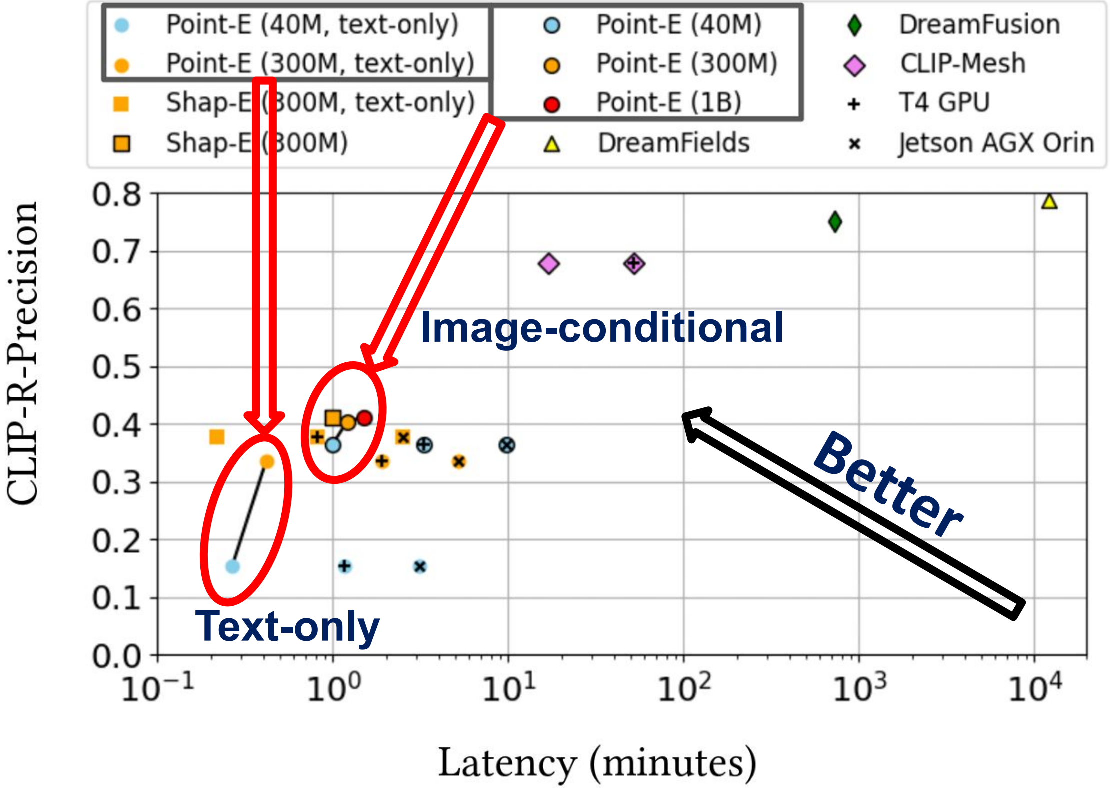
**Latency:**
**Text-only < Image-conditional**



# Latency-Quality Tradeoff

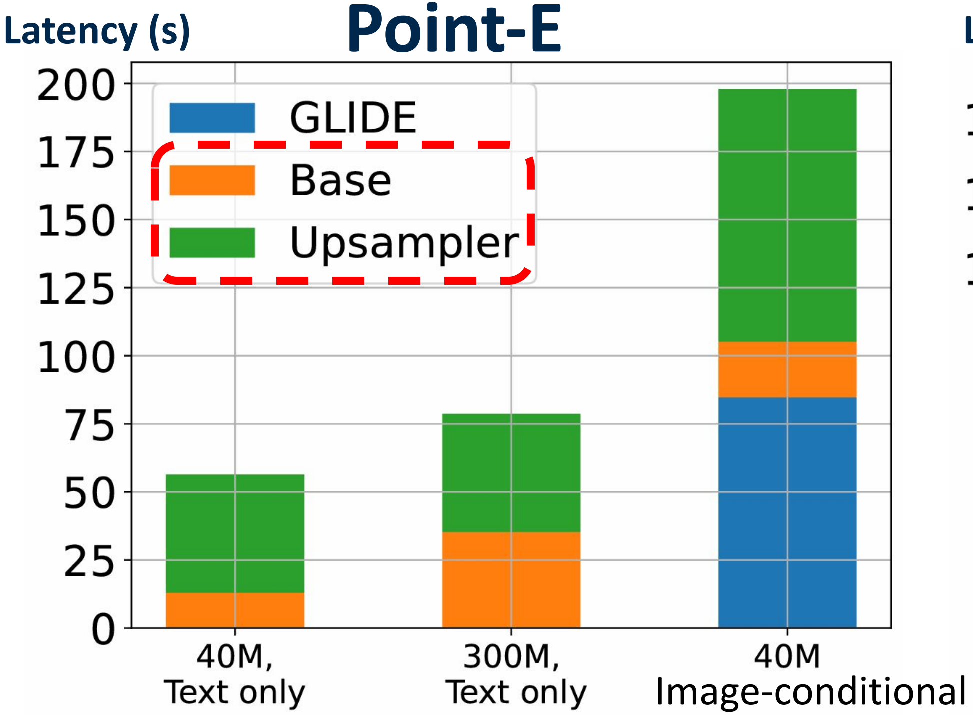
**Synthesis quality:**
**Image-conditional > Text-only**

**Latency:**
**Text-only < Image-conditional**

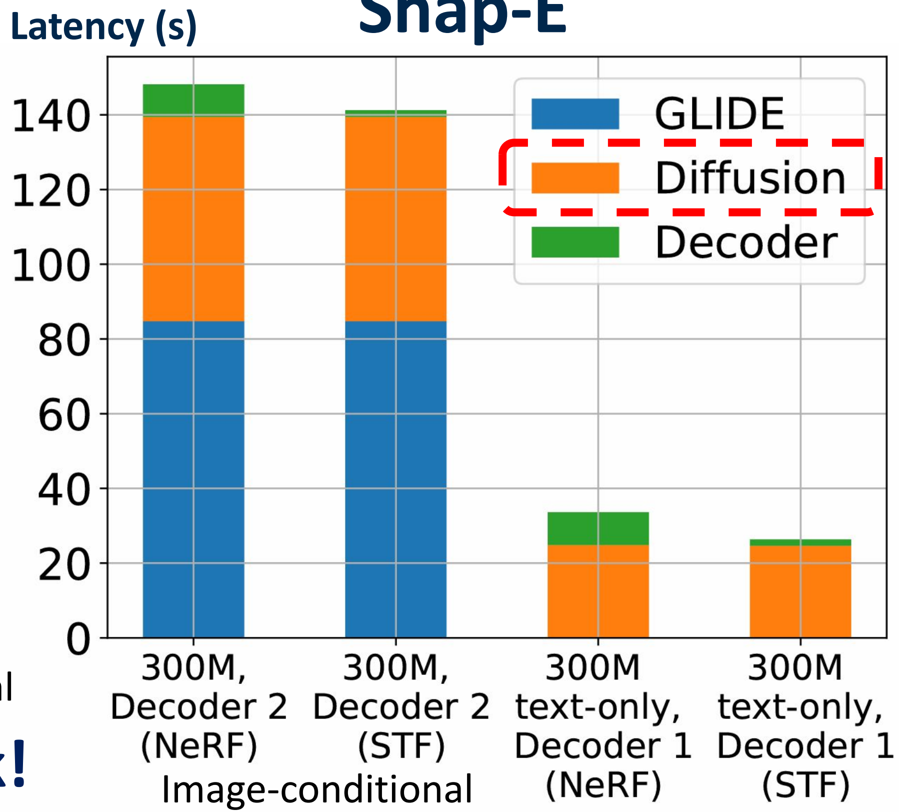


# Latency Breakdown

## Point-E



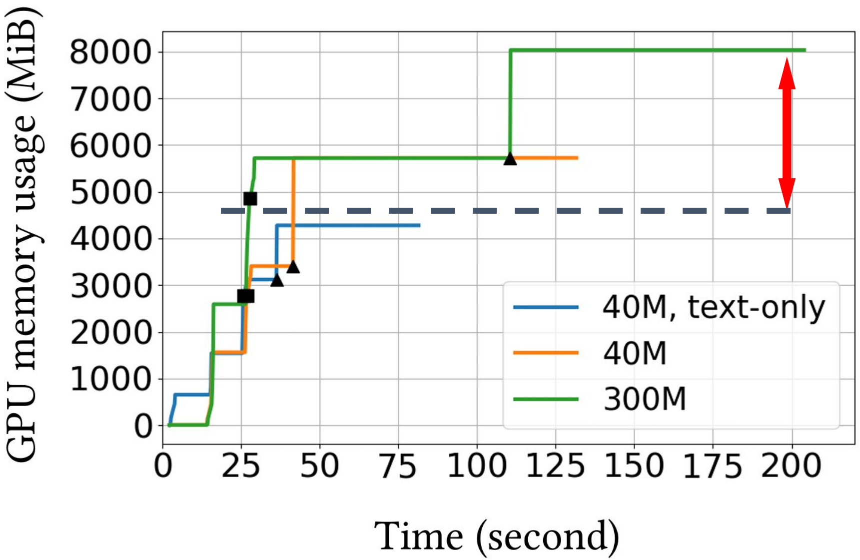

## Shap-E



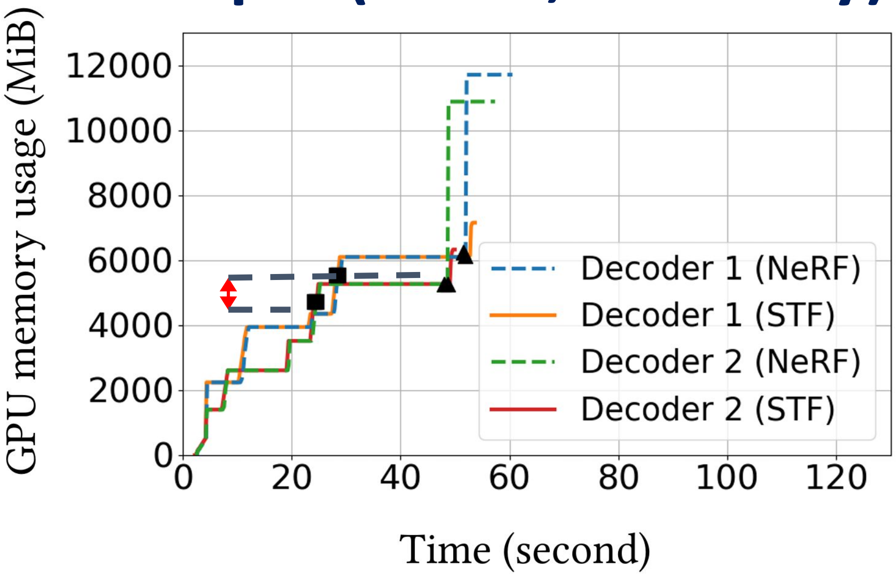

**Diffusion is a latency bottleneck!**

# GPU Memory Measurement

## Point-E




## Shap-E (300M, Text-only)




■ Start base diffusion ▲ Start upsampling

■ Start diffusion ▲ Start decoding (renderin

**Implicit representation can save memory usage during generation.**

# Model Optimization

## What to optimize?

## Diffusion process!

# Model Optimization

**How to optimize?**

- **Distillation**
- **Quantization**
- **Neural Architecture Search, Pruning, etc.**

# Model Optimization

## How to optimize?

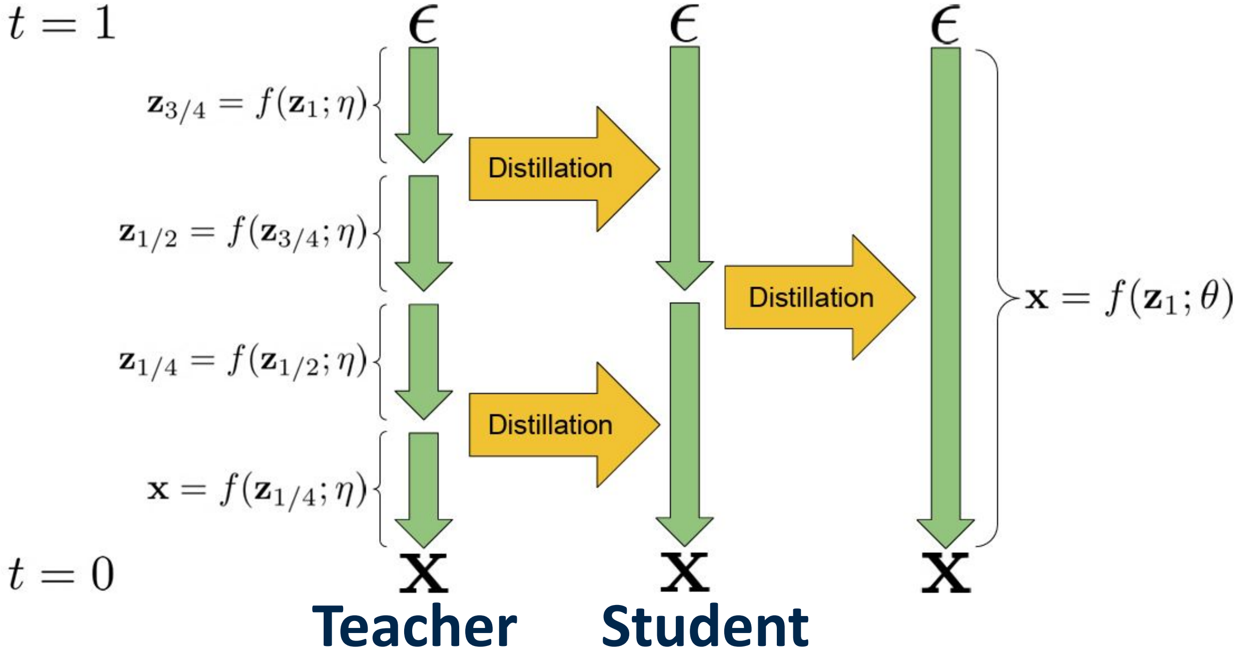
- Distillation
- Quantization

→ Can be generalized for other diffusion based models

- Neural Architecture Search, Pruning, etc.

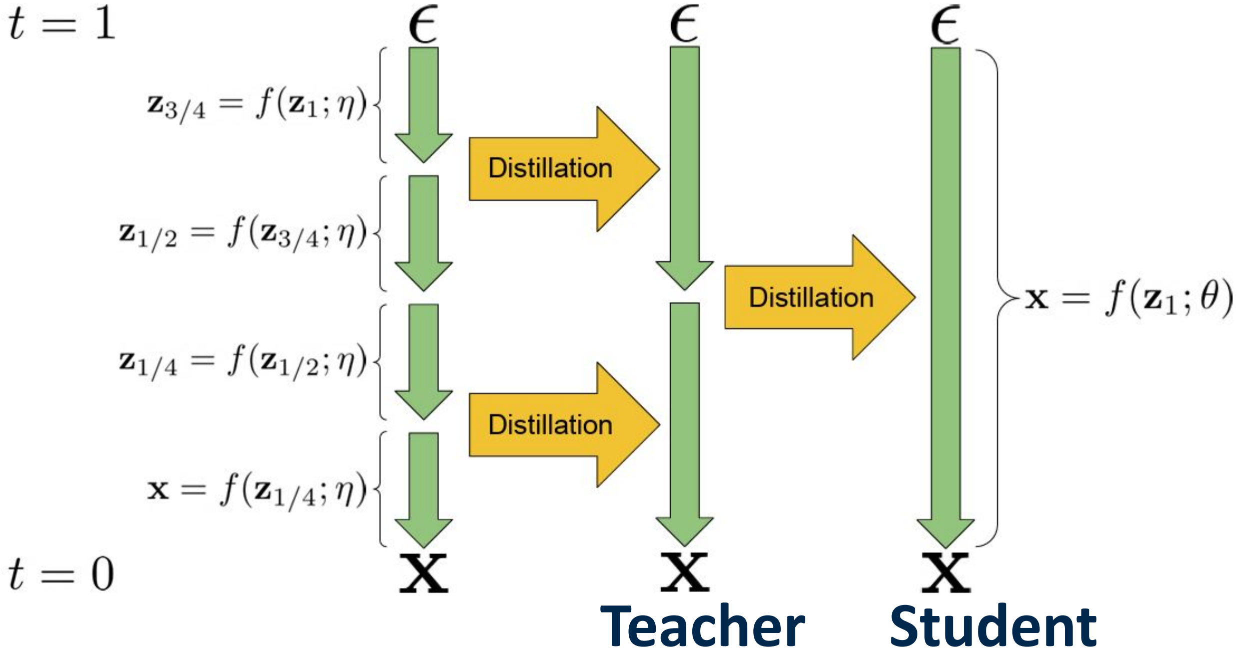
# Model Optimization: Distillation

## Speed up the model by reducing steps



# Model Optimization: Distillation

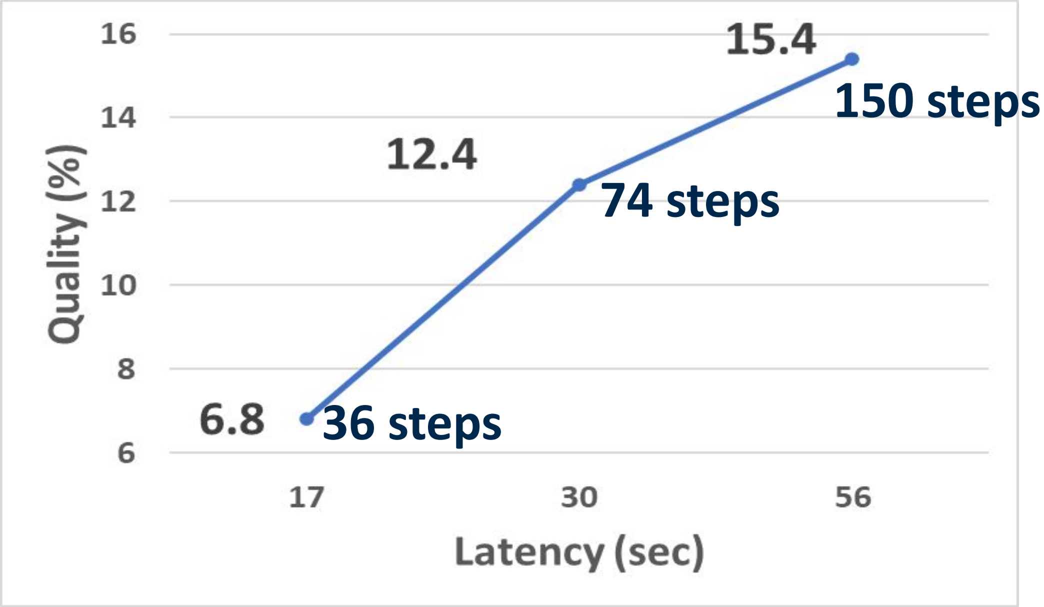
## Speed up the model by reducing steps



# Model Optimization: Distillation

## Speed up the model by reducing steps

Point-E results:




**Synthesis quality severely degrades at lower latency.**

# Model Optimization: Quantization

**Speed up** the model and **reduce memory usage** by using **lower precision parameters:** 32 bit → 8 bit

Quantization

**Point-E results:**

| Library | Layers | Quality ↑ | Speed |
|---|---|---|---|
| Original | n/a | 15.4% | ×1 |
| TensorRT | Linear | 10.2% | ×1.3 |
| TensorRT | All | 1.7% | ×1.8 |
| PyTorch (FBGEMM) | Linear | 11% | ×1.3 |

# Model Optimization: Quantization

**Speed up** the model and **reduce memory usage** by using **lower precision parameters:** 32 bit → 8 bit

Quantization

**Point-E results:**

| Library | Layers | Quality ↑ | Speed |
|---|---|---|---|
| Original | n/a | 15.4% | ×1 |
| TensorRT | Linear | 10.2% | ×1.3 |
| TensorRT | All | 1.7% | ×1.8 |
| PyTorch (FBGEMM) | Linear | 11% | ×1.3 |

**May need custom per-layer quantization**

# Summary

## Thank you! Questions?

Custom optimization (e.g. distillation, quantization) of text-to-3D models needed for mobile deployment.

Shap-E outperforms Point-E on mobile devices, possibly due to its efficient implicit representation.

Synthesis quality:

Text → 2D → 3D > Text → 3D