BOSTON UNIVERSITY

Deep Learning for Data Science
DS 542

https://dl4ds.github.io/fa2025/

Using Pre-trained Models

Original slides by Tom Gardos,
other content from Understanding Deep Learning unless otherwise cited

Plan for Today

Using pre-trained models

- Model embeddings
- Classifier-driven generation
- ControlNet

Rest of Semester

- Data preparation and augmentation
- Reasoning and world models
- Graphical neural networks
- Deep reinforcement learning

Large Pre-Trained Models

- ImageNet models
- Large language models – base models
- Image generation models (e.g. Stable Diffusion)
- Time series foundation models (Google's TimesFM)
- Robot foundation models (in progress!)

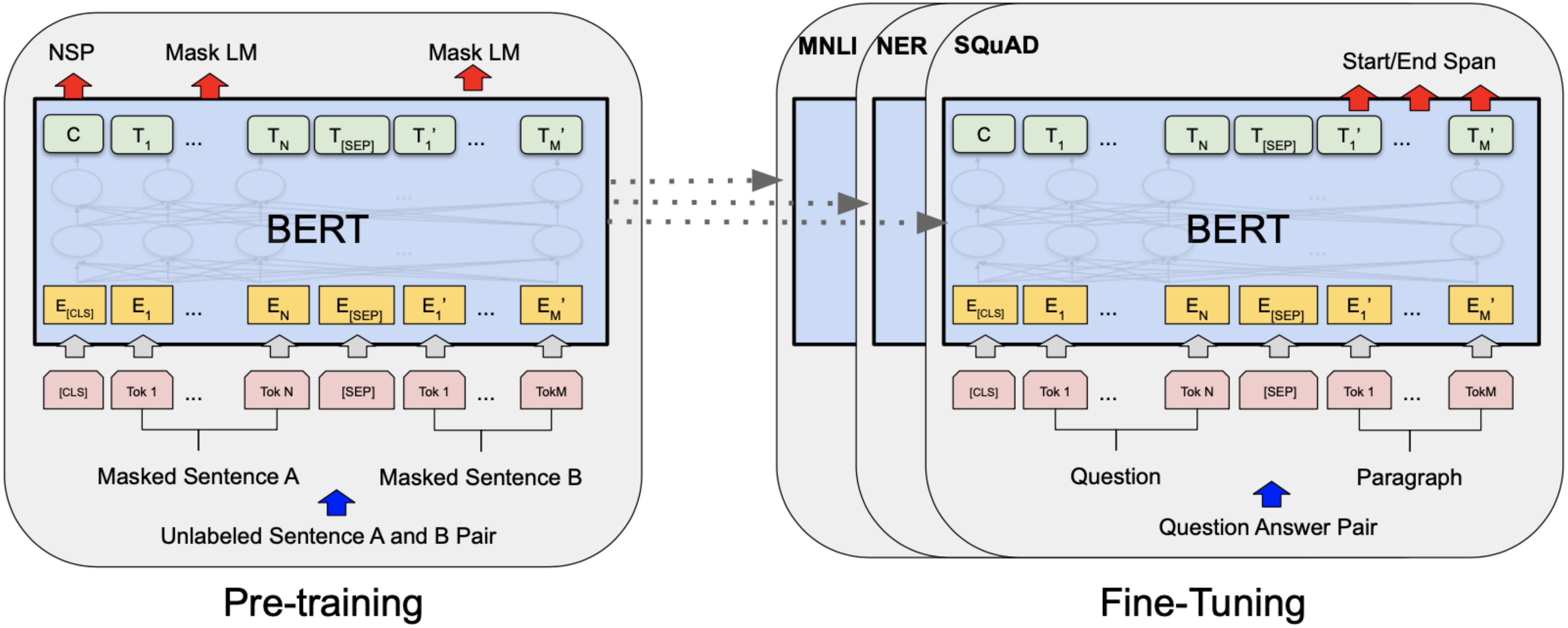
Typical Pre-Trained Model Built for One Task

- But nowadays, expected to be tweaked in different directions.

- For example, all the post-training we discussed for language models.
  - Or fine-tuning for different problems.

- “Foundation” label represents intent/hope to build on top of these models.
  - Amortize the big expense of training huge model over many applications.

Repurposing for classification



Full or Feature-based Fine Tuning?

Full fine tuning

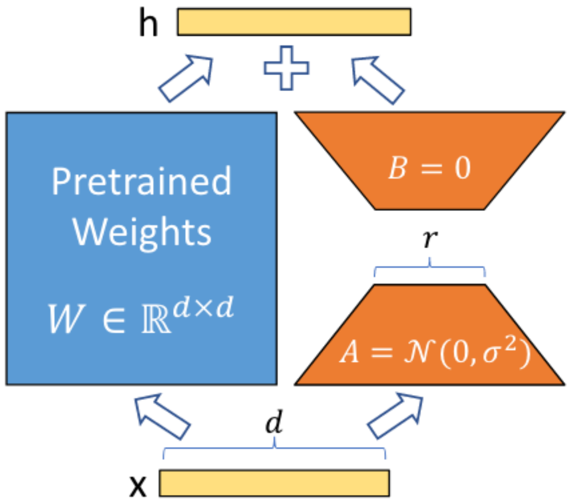
- Any parameter in the original model can change.
- Maximally adaptive to new problem.
- Full set of new weights to store.
- Medium expensive to train, but cheaper than training a new model from scratch.

Feature-based fine tuning

- Changes limited to new layer(s) after frozen weights.
- Flexibility usually limited to linear model or similar.
- Small set of new weights to store - usually just one layer.
- Very cheap to train. Original model just needs to generate features.

Low Rank Adaptations (LoRA)

- Compromise between full fine-tuning and feature-based classification.
- Fine-tunes all weight matrices, but with constrained changes.
- Sometimes applied to the original problem too.
  - Used to encourage certain styles or subjects.


h
Pretrained Weights
$W \in \mathbb{R}^{d \times d}$
$B = 0$
r
$A = \mathcal{N}(0, \sigma^2)$
d
x

Embeddings as representations

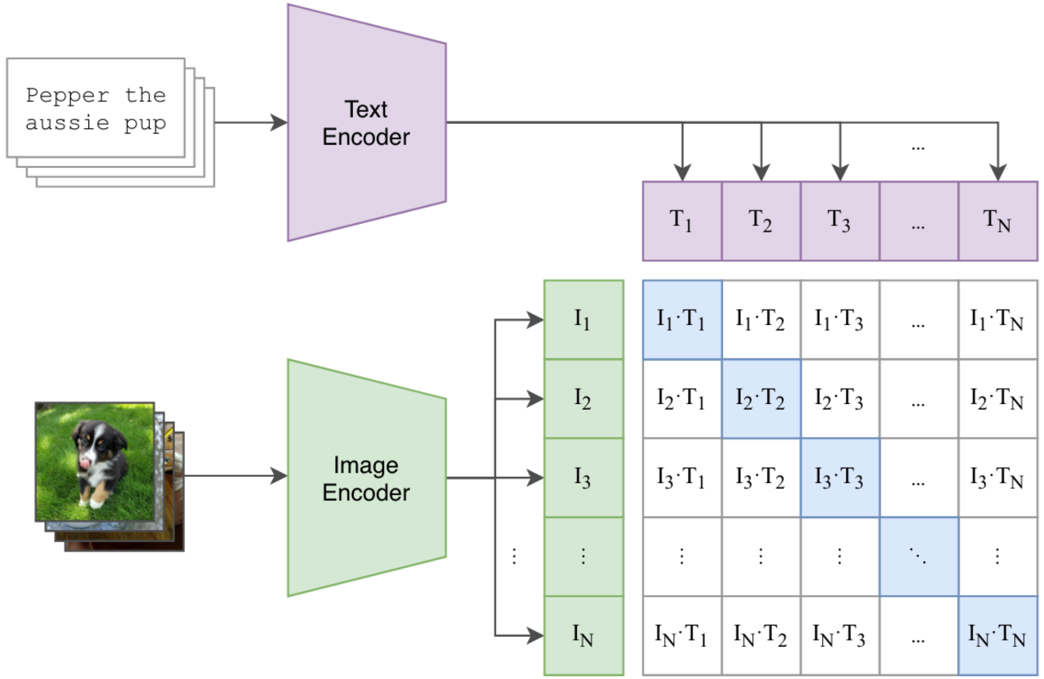
- Previously took these embedded vectors from hidden layers and used them as feature vectors.

- Can we use them as representations?
  - Embedding vector instead of raw text or image pixels?
  - Are embedding vectors of similar text or images similar in vector space?
  - Are embedding vectors useful document vectors?

Contrastive Language Image Pretraining (2021)

- Previously saw deliberate training of text and image encodings to work together.
- Good models tend to have some of these properties already…

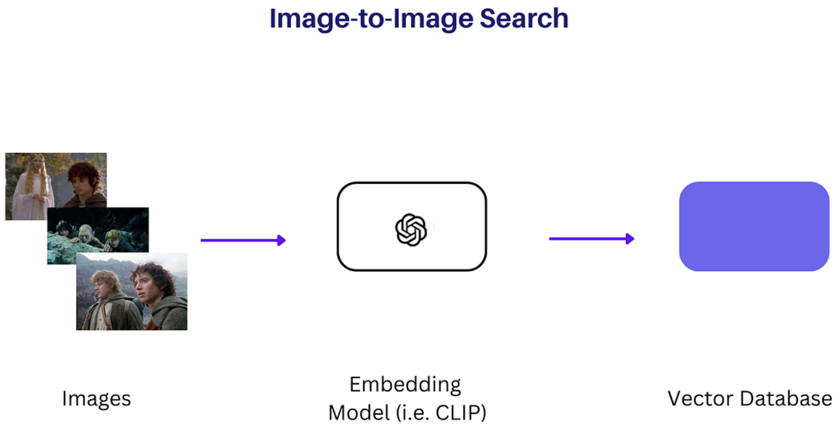
(1) Contrastive pre-training



	T_1	T_2	T_3	...	T_N
I_1	$I_1 \cdot T_1$	$I_1 \cdot T_2$	$I_1 \cdot T_3$	...	$I_1 \cdot T_N$
I_2	$I_2 \cdot T_1$	$I_2 \cdot T_2$	$I_2 \cdot T_3$	...	$I_2 \cdot T_N$
I_3	$I_3 \cdot T_1$	$I_3 \cdot T_2$	$I_3 \cdot T_3$	...	$I_3 \cdot T_N$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
I_N	$I_N \cdot T_1$	$I_N \cdot T_2$	$I_N \cdot T_3$	...	$I_N \cdot T_N$

Embeddings as Search Keys

- Searching for similar images works with CLIP embeddings.
- Searching for similar text with language model embeddings usually works too.



https://medium.com/@tenyks_blogger/how-to-build-an-image-to-image-search-tool-using-clip-pinecone-b7b70c44faac

Retrieval Augmented Generation

Idea:

1. Use document embeddings to lookup relevant context.
2. Include documents into large language model prompts.

Specifically,

1. Pre-compute and store embeddings for all documents.
2. Compute embedding for each query.
3. Lookup documents with closest embeddings to query.
4. Combine documents and query into one long LLM prompt.

Retrieval Augmented Generation Example

I need a question answered based on our internal company policies.

Answer it only based on the following documents.

Document 1:

Blah blah blah

Document 2:

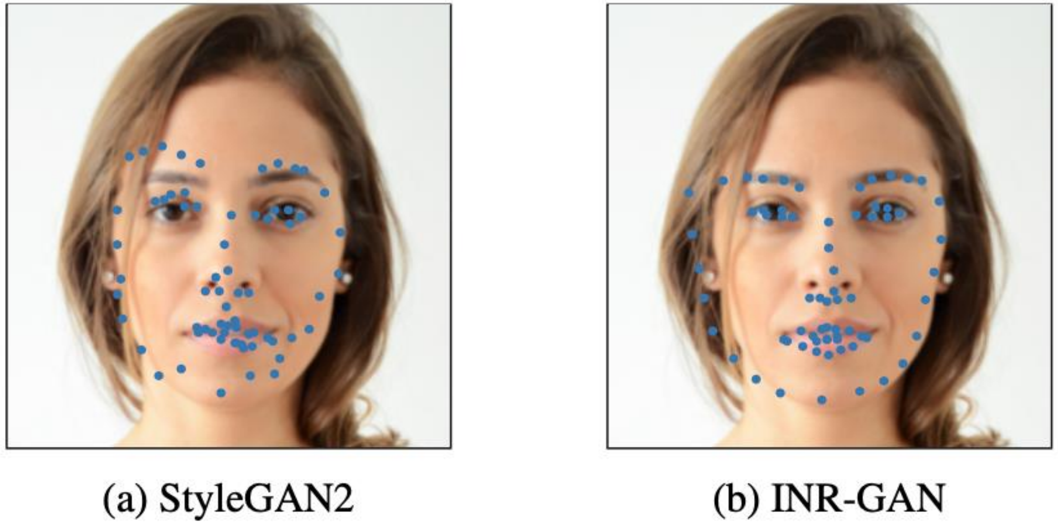
Blah blah blah

Question: Are we allowed to play foosball during working hours?

Good Latent Codes Work Similarly

- Many of the claims just now about embeddings from pre-trained models also apply to good latent codes.
- The key points on the right were computed as linear functions of the latent codes.
- Seems to correlated with being well-behaved?

Image source: Adversarial Generation of Continuous Images (Skorokhodov et al, 2021)



(a) StyleGAN2

(b) INR-GAN

Any Questions?

???

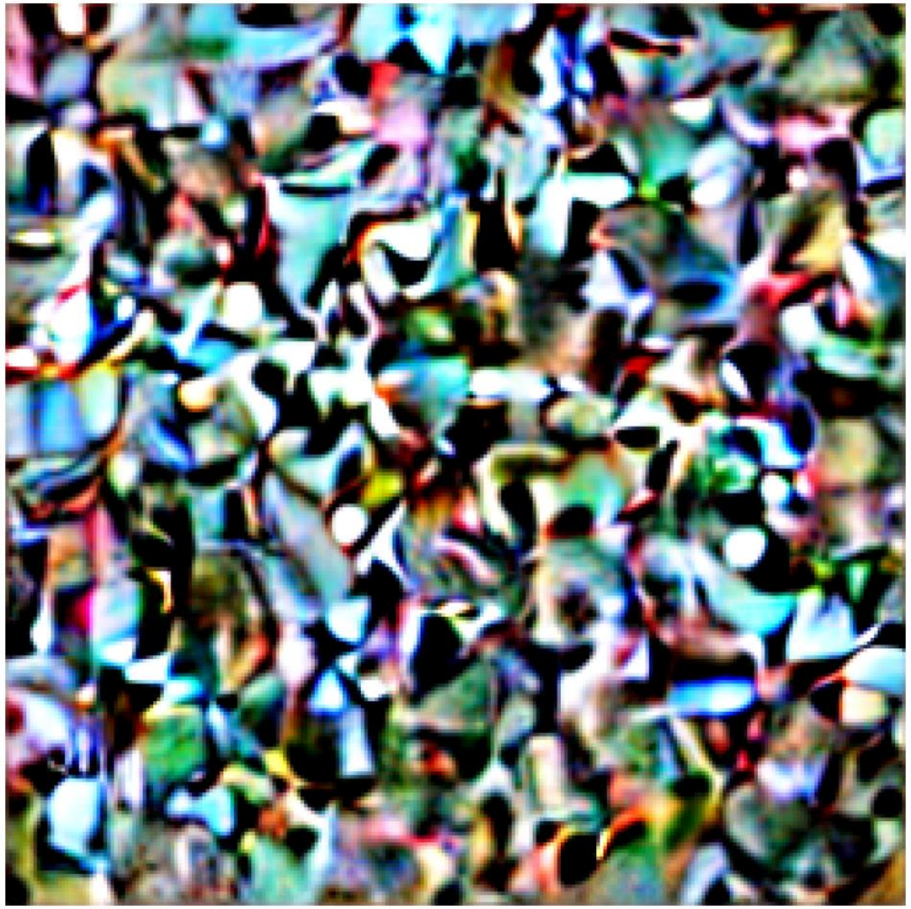
Moving on

- Model embeddings
- Classifier-driven generation
- ControlNet

Previously in Homework 2…

- Problem: morph this into a target image
  - Gradient descent is a general, but slow way to invert models.

- More interesting:
  - Create a model for a given class.
  - Easy if model was designed with classifiers in mind and/or a shared text/image latent space.



Gradient Descent to Generate a Class?

- If latent space is not well-behaved, you are probably going to reinvent adversarial inputs.

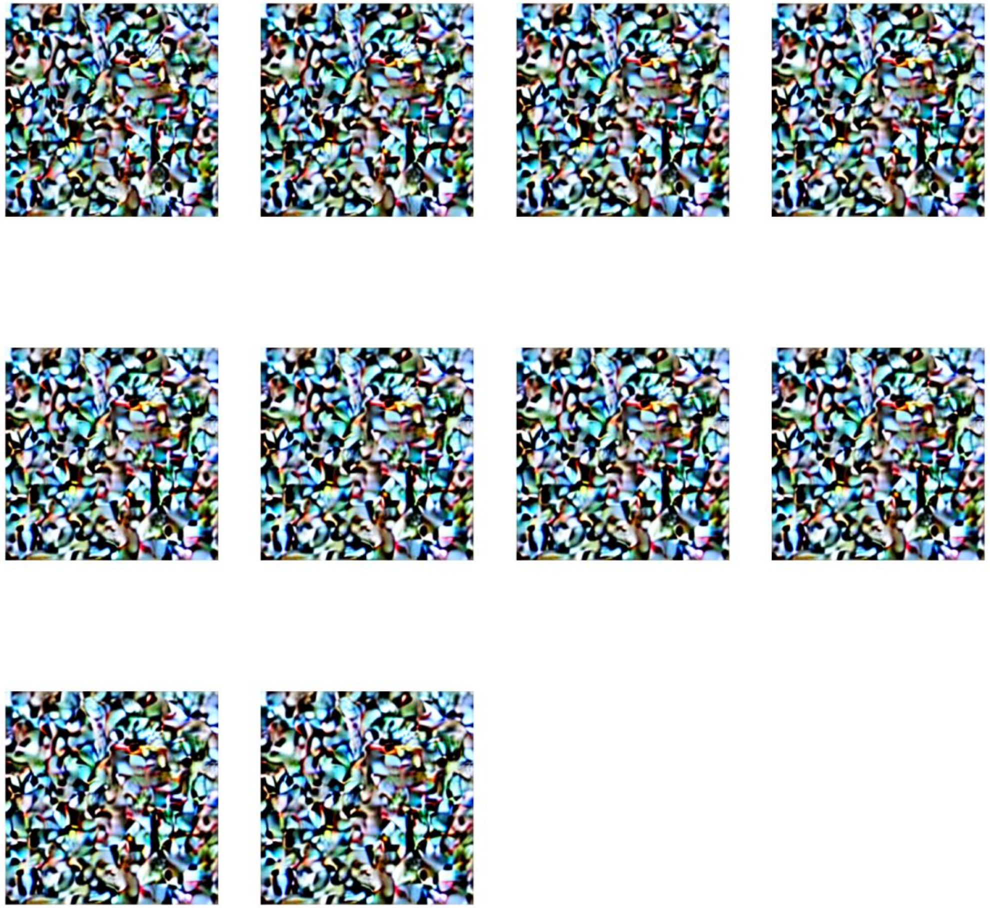


Image is from 100 steps optimizing random Stable Diffusion latent for ImageNet class “chocolate sauce”.

- Loss 0.15682560205459595

Can it be made to work?

- Yes, but janky.

- For diffusion models,
  - Usually works by integrating gradient steps to improve class probability and diffusion steps to get a decent image out.
  - Often used as a strawman before presenting an integrated version.
  - You can implement if you can trigger individual diffusion steps.

Diffusion Models Beat GANs on Image Synthesis (Dhariwal et al, 2021)

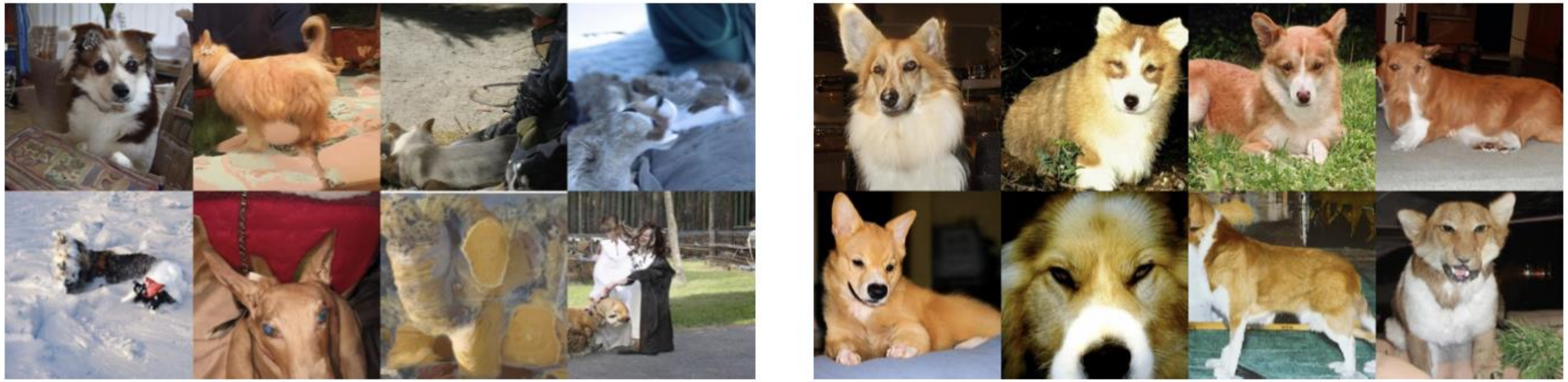


Figure 2: Samples from an unconditional diffusion model with classifier guidance to condition on the class "Pembroke Welsh corgi". Using classifier scale 1.0 (left; FID: 33.0) does not produce convincing samples in this class, whereas classifier scale 10.0 (right; FID: 12.0) produces much more class-consistent images.

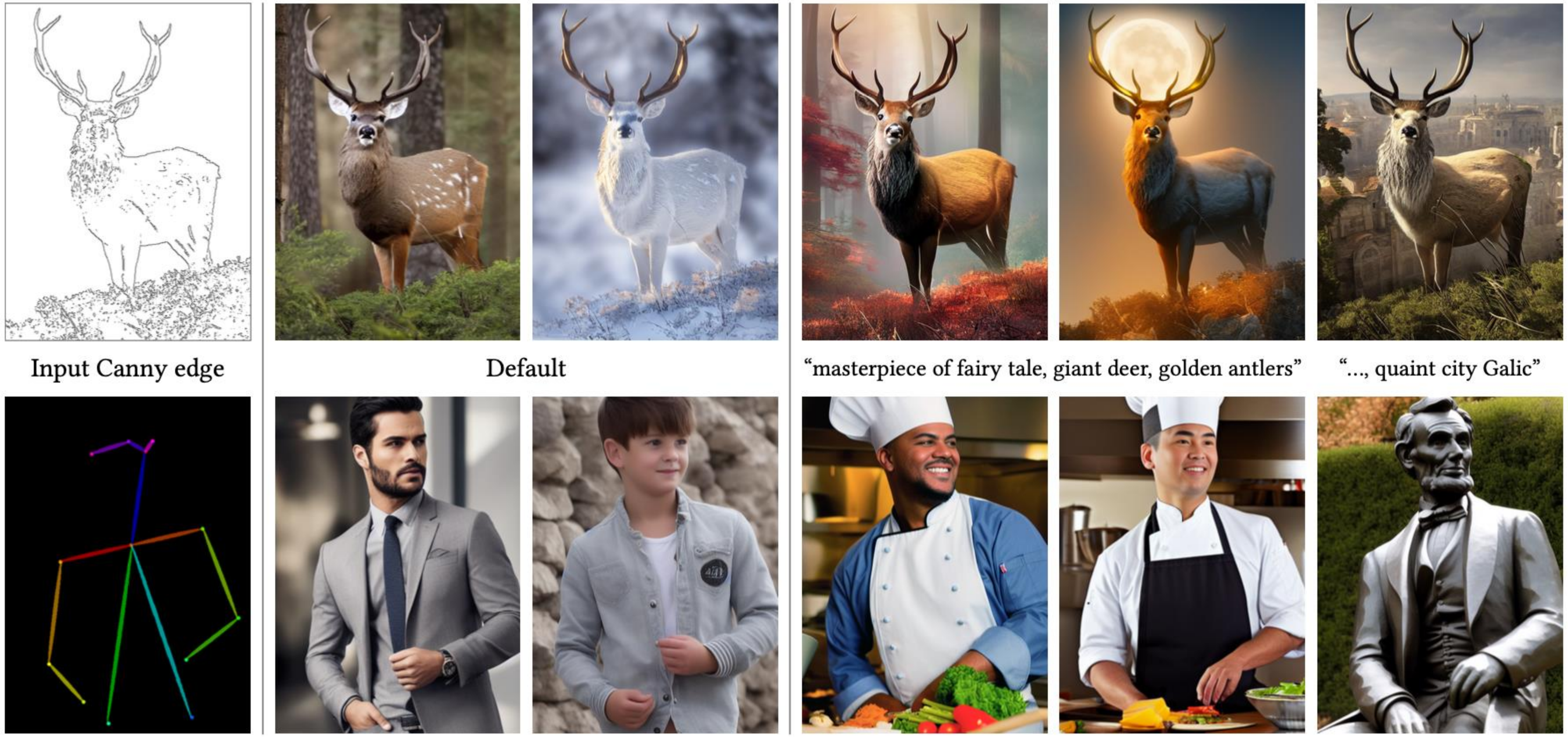
Any Questions?

???

Moving on

- Model embeddings
- Classifier-driven generation
- ControlNet

Adding Conditional Control to Text-to-Image Diffusion Models (Zhang et al, 2023)



Input Canny edge

Default

“masterpiece of fairy tale, giant deer, golden antlers”

“..., quaint city Galic”

Input human pose

Default

“chef in kitchen”

“Lincoln statue”

A Computational Approach to Edge Detection (Canny, 1986)

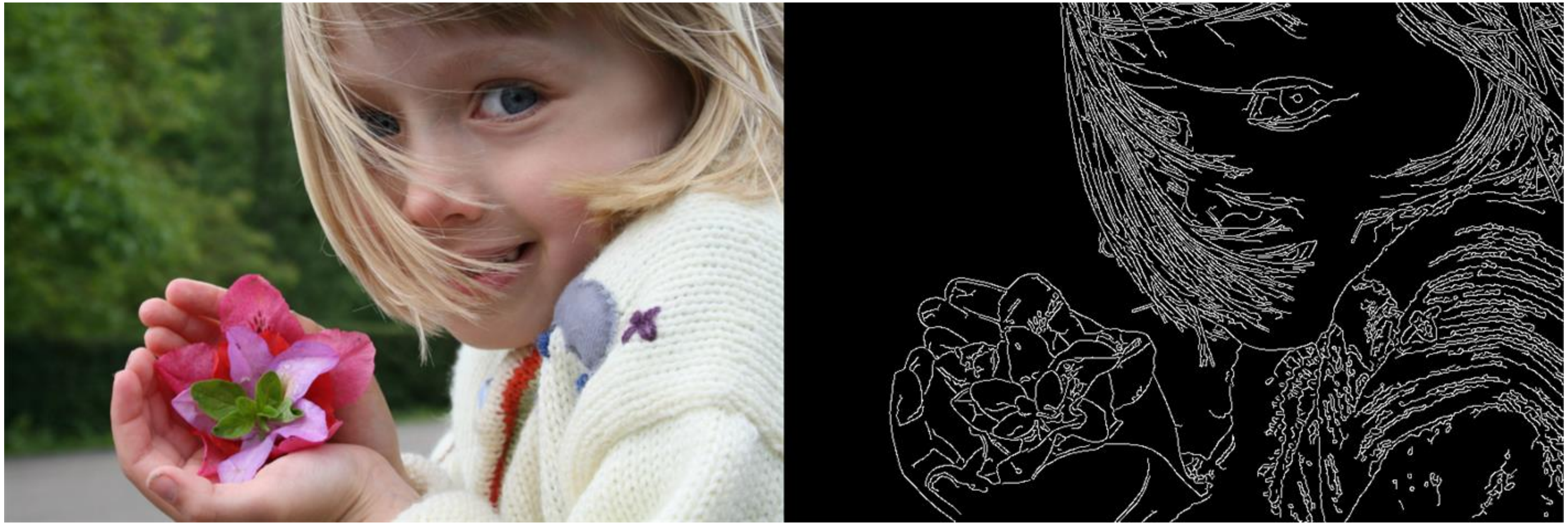
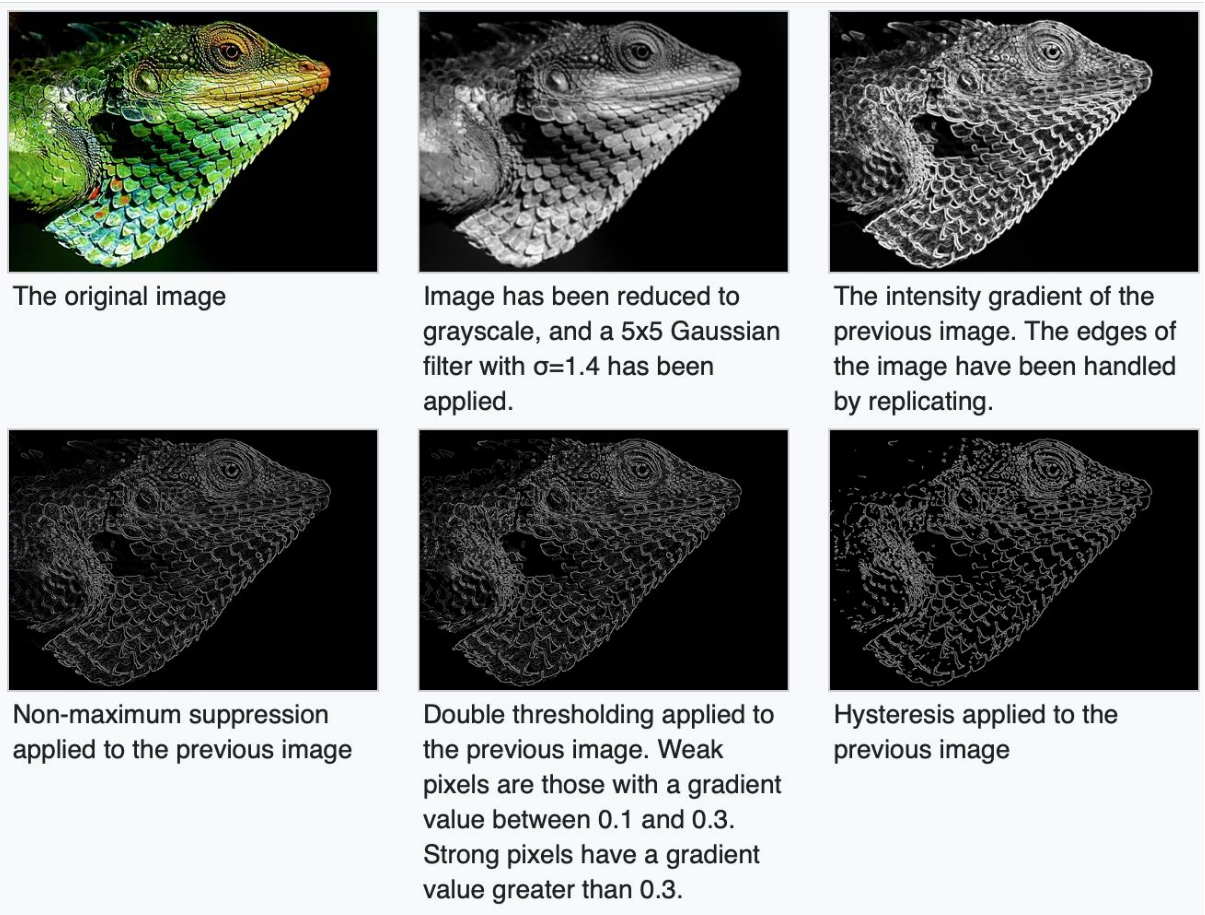


Image source: https://en.wikipedia.org/wiki/Canny_edge_detector

A Computational Approach to Edge Detection (Canny, 1986)



The original image

Image has been reduced to grayscale, and a 5x5 Gaussian filter with σ=1.4 has been applied.

The intensity gradient of the previous image. The edges of the image have been handled by replicating.

Non-maximum suppression applied to the previous image

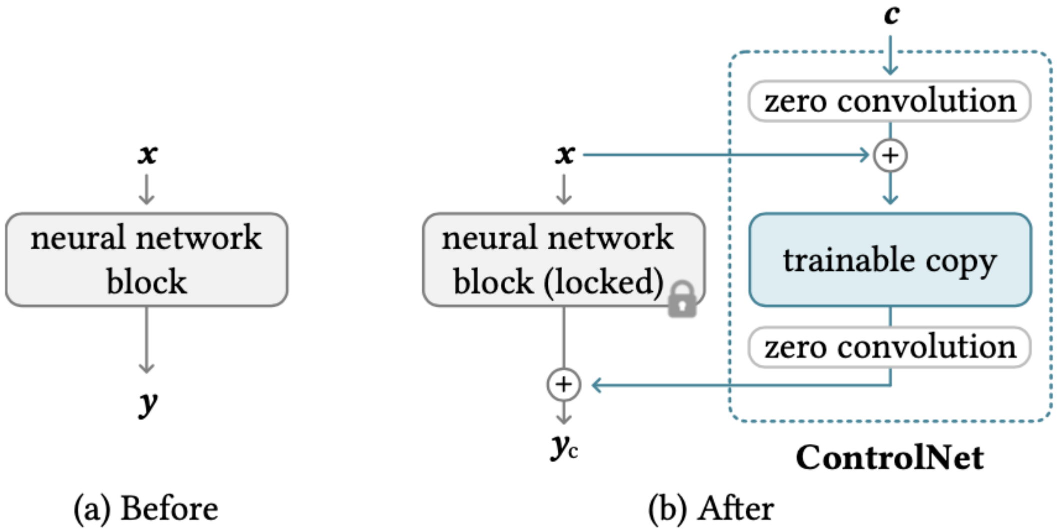
Double thresholding applied to the previous image. Weak pixels are those with a gradient value between 0.1 and 0.3. Strong pixels have a gradient value greater than 0.3.

Hysteresis applied to the previous image

Image source: https://en.wikipedia.org/wiki/Canny_edge_detector

ControlNet Proposal

- Use a pre-trained text-to-image diffusion model (text optional).
- Build new image-to-image model.
    - Takes in conditioning input (e.g. Canny edges)
    - Generates new image matching that conditioning.
- New architecture consists of:
    - Frozen copy of original image model
    - New trainable copy of the model
    - Connected by "zero convolution" layers.

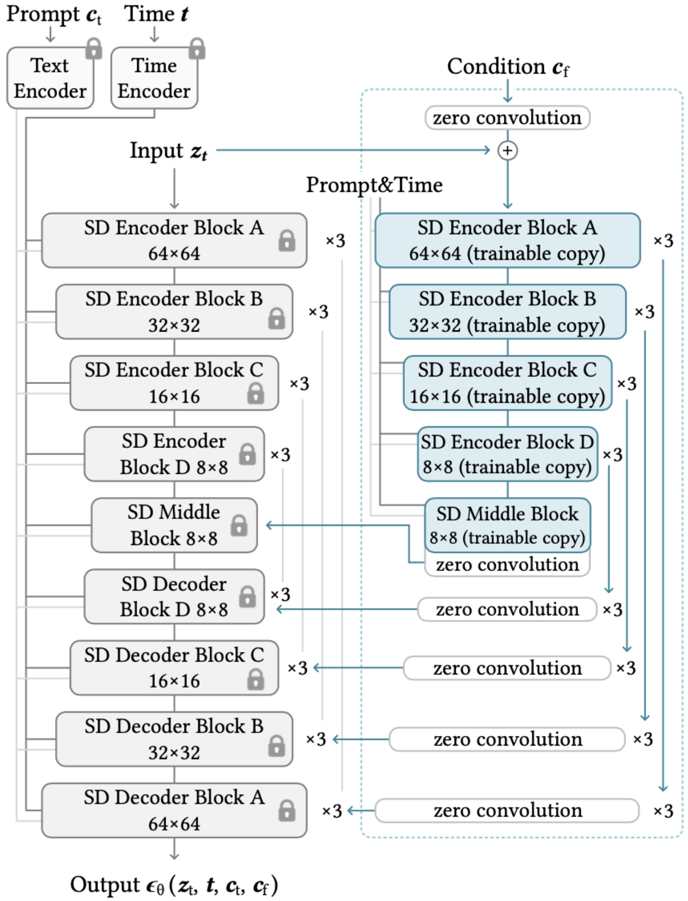


Why This Design?

- Frozen copy of original model protects against catastrophic forgetting.

- Zero convolutions protect against harmful noise early on.
    - If trainable copy is helpful, then the weights can change from zero.
    - Weights will stay close to zero until consistently helpful.

ControlNet for Stable Diffusion



- Input condition uses randomly initialized convolutions to transform from image size to latent size.
- ControlNet structure applied to copy encoder blocks.
- Copied encoder blocks are only wired into the residual connections on the decoder blocks.

ControlNet Training

- Similar to standard diffusion training.

$$\mathcal{L} = \mathbb{E}_{z_0, t, c_t, c_f, \epsilon \sim \mathcal{N}(0,1)} \left[\left\| \epsilon - \epsilon_\theta(z_t, t, c_t, c_f)) \right\|_2^2 \right]$$

With extra conditioning c_f added…

- This formula includes text conditioning c_t (previously covered).
- During training, c_t is set to 0 vector 50% of the time to emphasize adaptations based on c_f.

Adding Conditional Control to Text-to-Image Diffusion Models (Zhang et al, 2023)

ControlNet – Sudden Convergence

According to the authors…

- Training takes a while, but output quality is high because of the zero convolutions.
- “We observe that the model does not gradually learn the control conditions but abruptly succeeds in following the input conditioning image; usually in less than 10K optimization steps.”

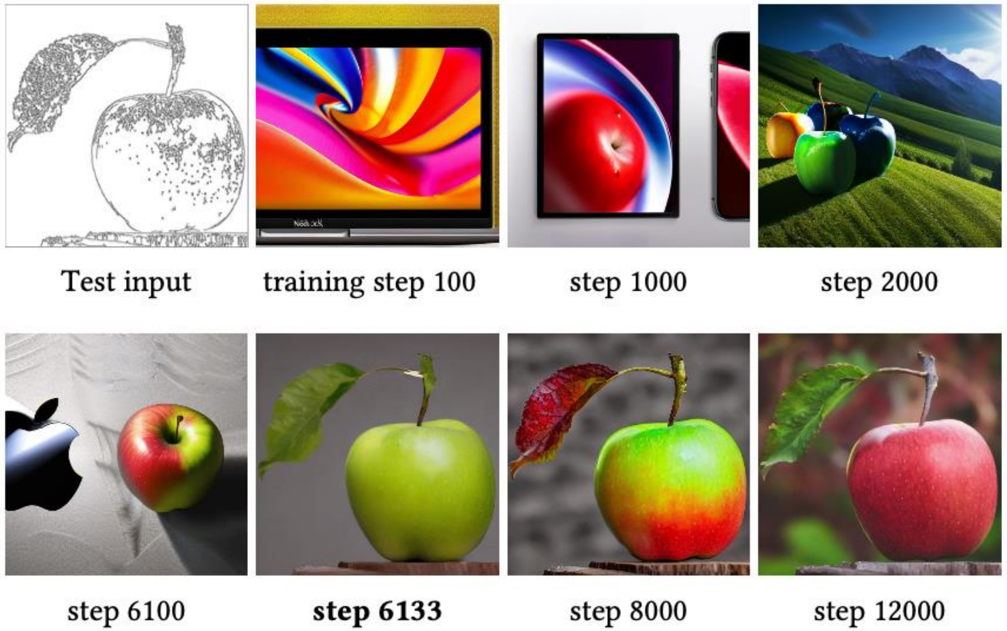


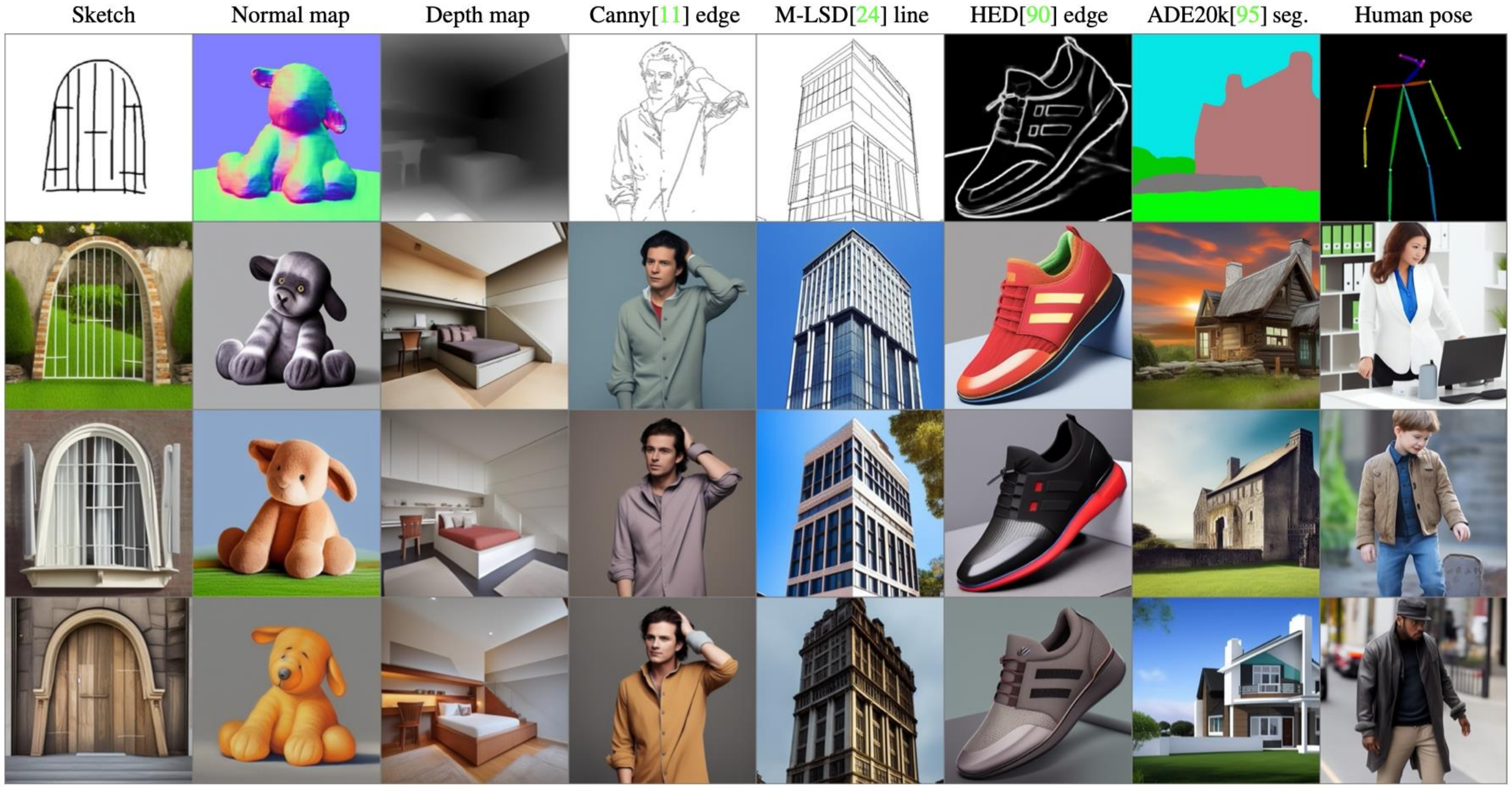
Figure 4: The sudden convergence phenomenon. Due to the zero convolutions, ControlNet always predicts high-quality images during the entire training. At a certain step in the training process (*e.g.*, the 6133 steps marked in bold), the model suddenly learns to follow the input condition.

Adding Conditional Control to Text-to-Image Diffusion Models (Zhang et al, 2023)

What is Happening During Training?

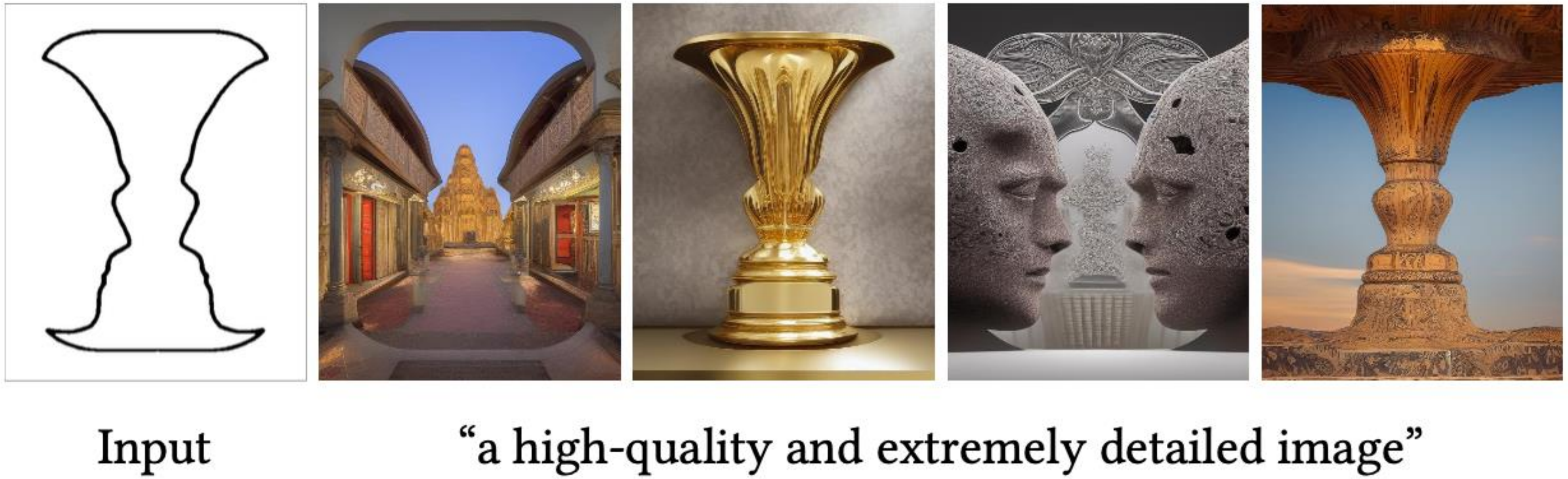
???

Controlling Stable Diffusion without Prompts



Adding Conditional Control to Text-to-Image Diffusion Models (Zhang et al, 2023)

Ambiguous Shapes



Adding Conditional Control to Text-to-Image Diffusion Models (Zhang et al, 2023)

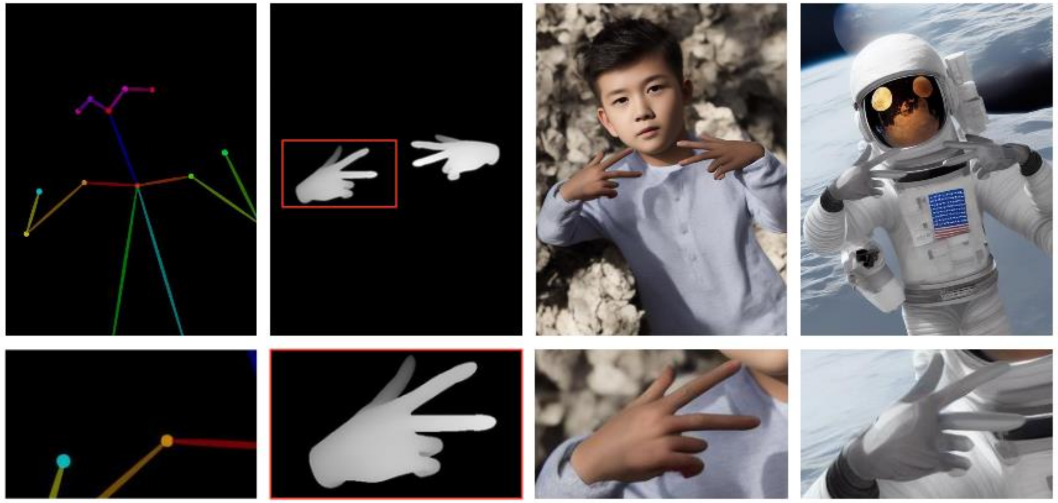
ControlNet Conditioning

- Canny edges
- Hough lines
- user scribbles
- human key points
- segmentation maps
- shape normal
- Depths
- cartoon line drawings

Adding Conditional Control to Text-to-Image Diffusion Models (Zhang et al, 2023)

Combining ControlNets

- Just add outputs of both ControlNet modules!

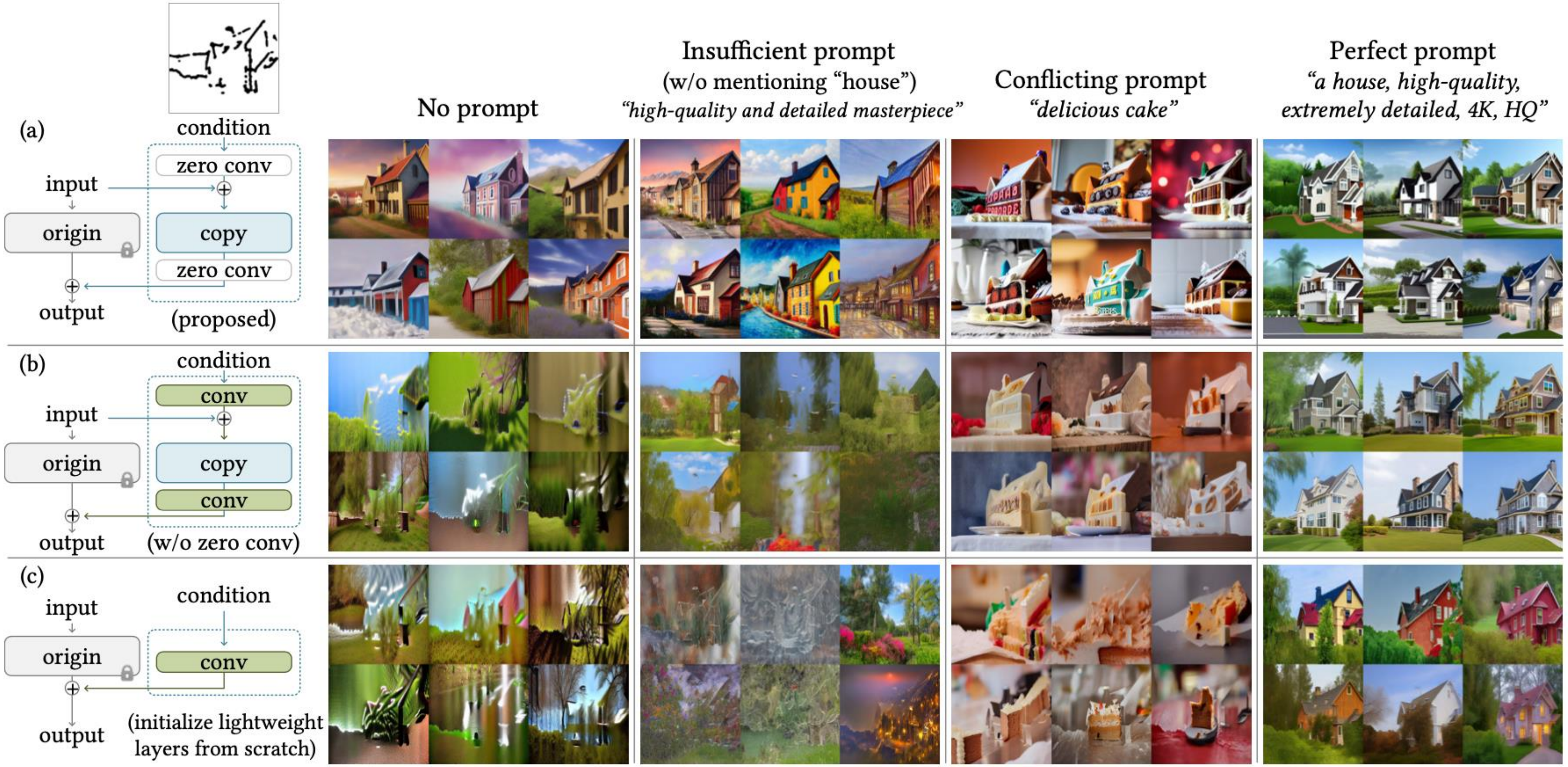


Multiple condition (pose&depth) “boy” “astronaut”

Figure 6: Composition of multiple conditions. We present the application to use depth and pose simultaneously.

Adding Conditional Control to Text-to-Image Diffusion Models (Zhang et al, 2023)

ControlNet Ablation Study



Adding Conditional Control to Text-to-Image Diffusion Models (Zhang et al, 2023)

Any Questions?

???

Moving on

- Model embeddings
- Classifier-driven generation
- ControlNet