Module 2: Unsupervised Representation Learning & Generative Modeling



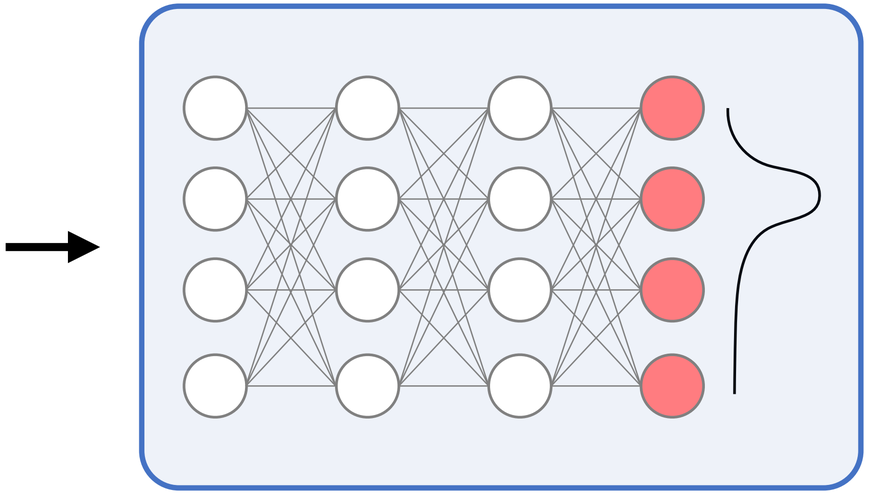
Prof. Vincent Sitzmann

What is Representation Learning?

A historical perspective

Image Classification

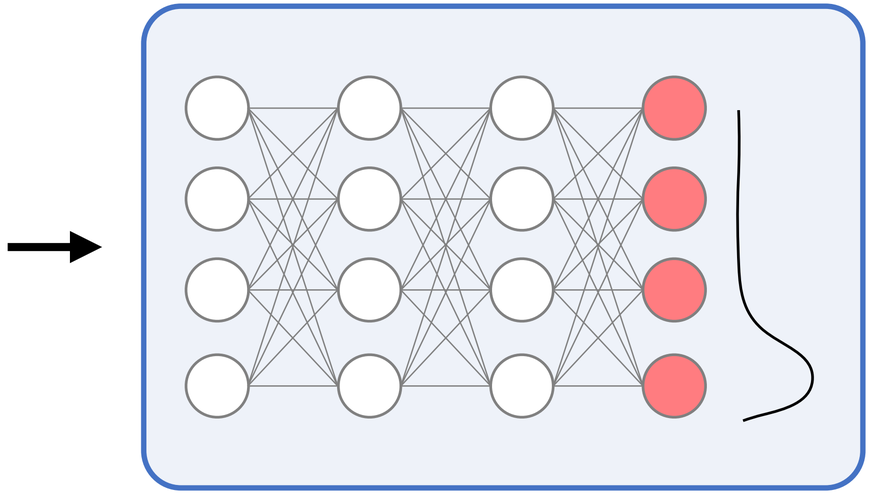




“Guitar”

Image Classification

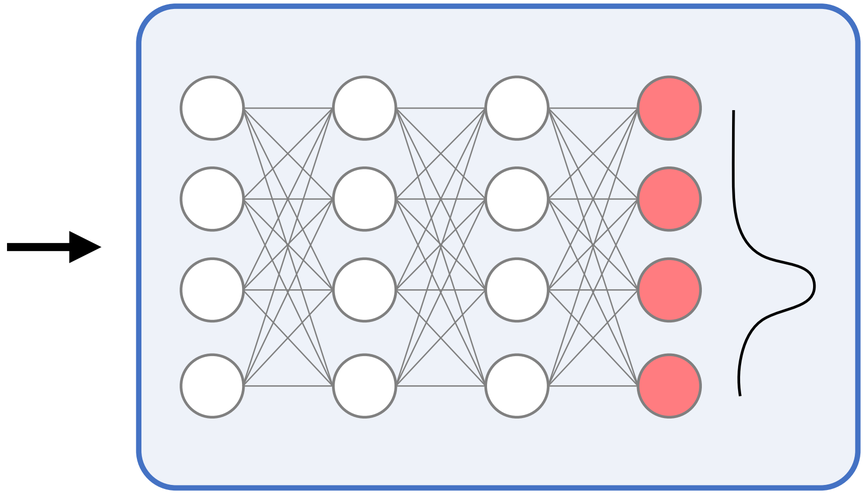




“Dog”

Image Classification





“Car”

New Task: Image Segmentation

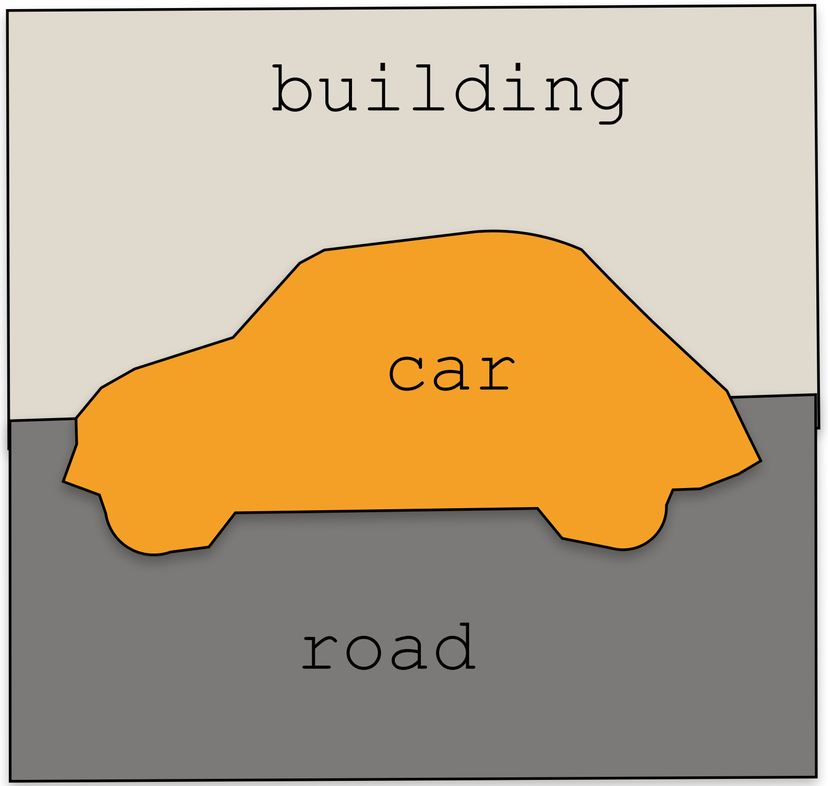




New Task: Image Segmentation



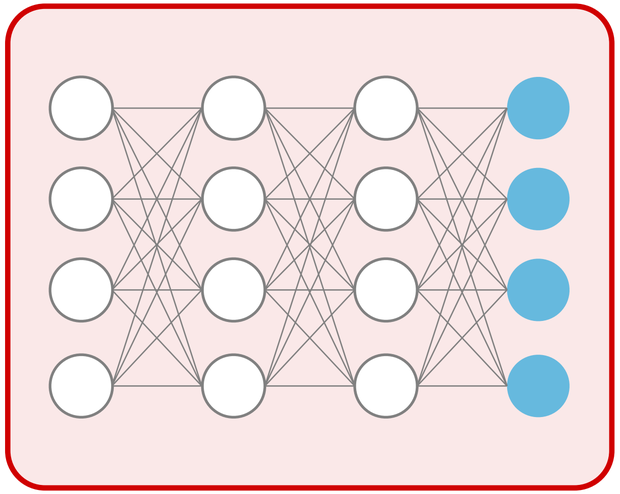


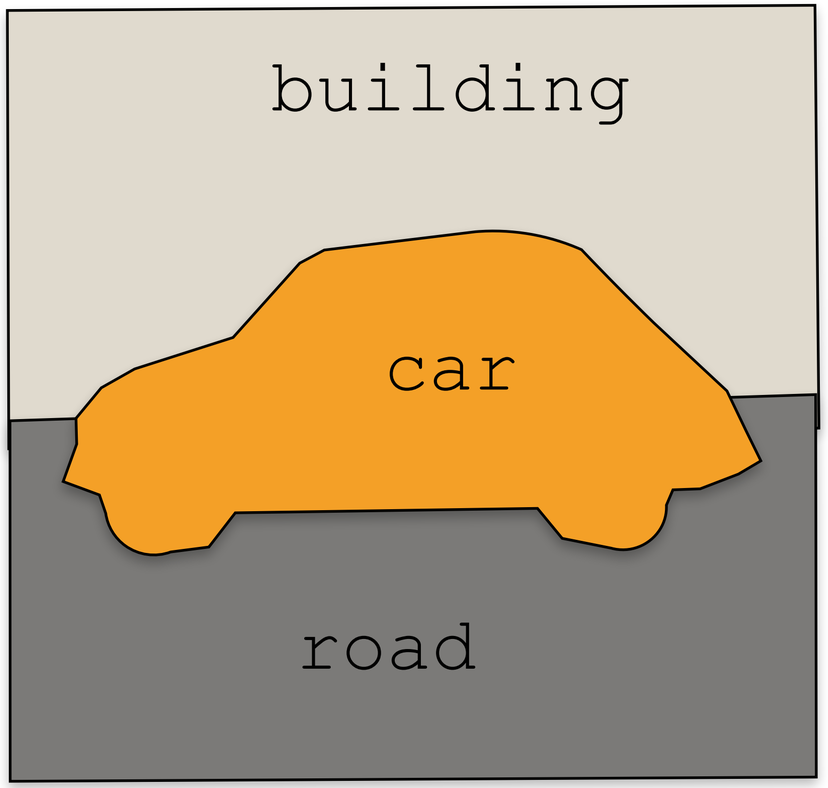


New Task: Image Segmentation



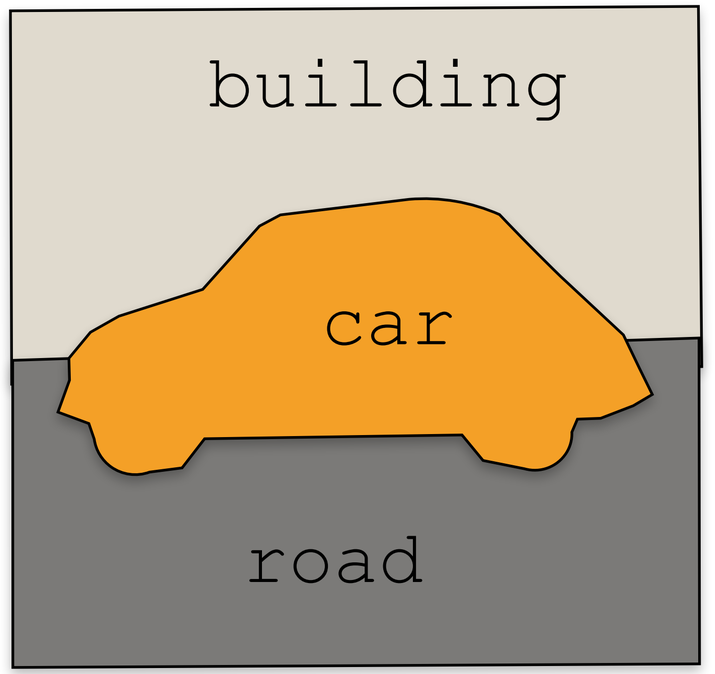
We *could* train a *new* model from scratch…





Re-use **Representation Learned by Pre-Trained Model**

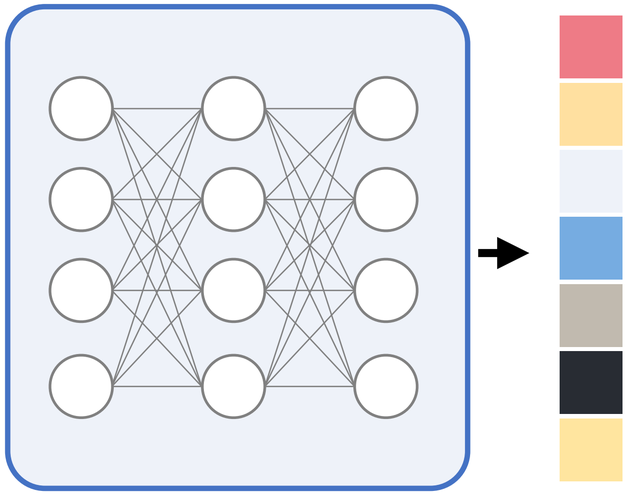


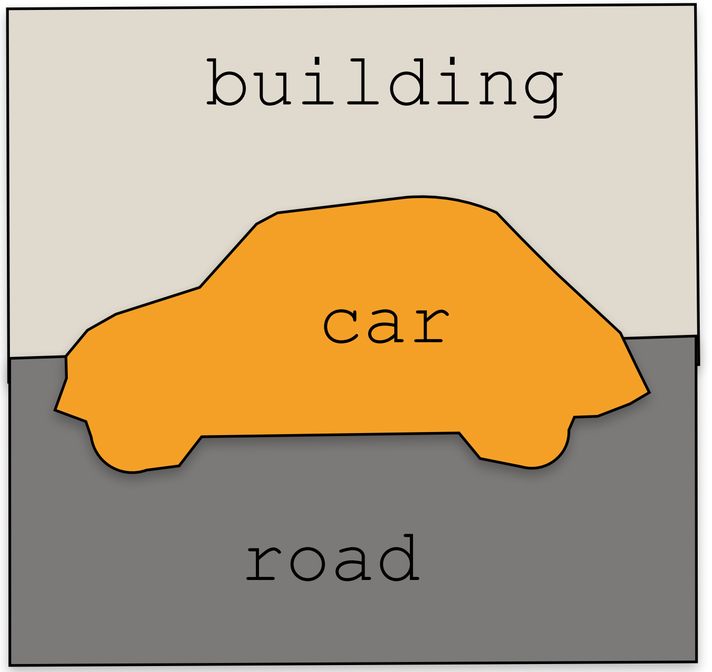


Re-use **Representation Learned by Pre-Trained Model**

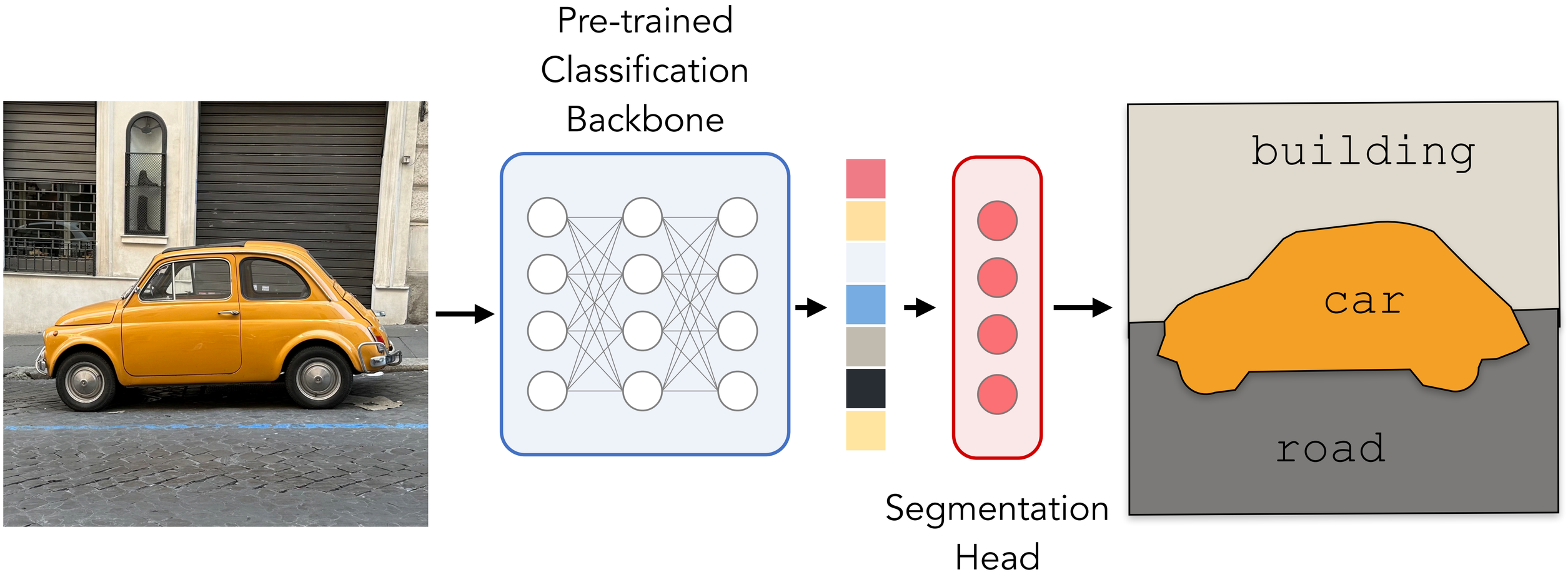


Pre-trained
Classification
Backbone





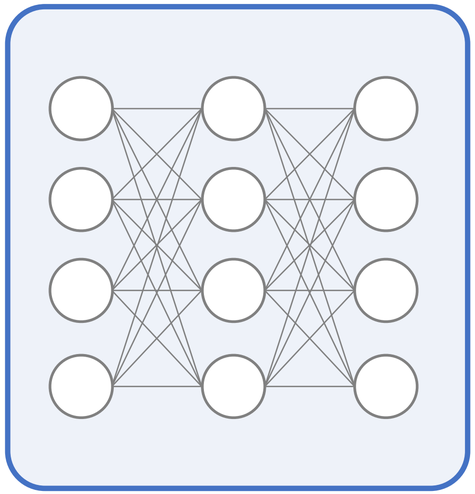
Re-use **Representation Learned by Pre-Trained Model**



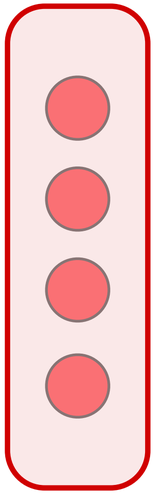
Re-use **Representation Learned by Pre-Trained Model**



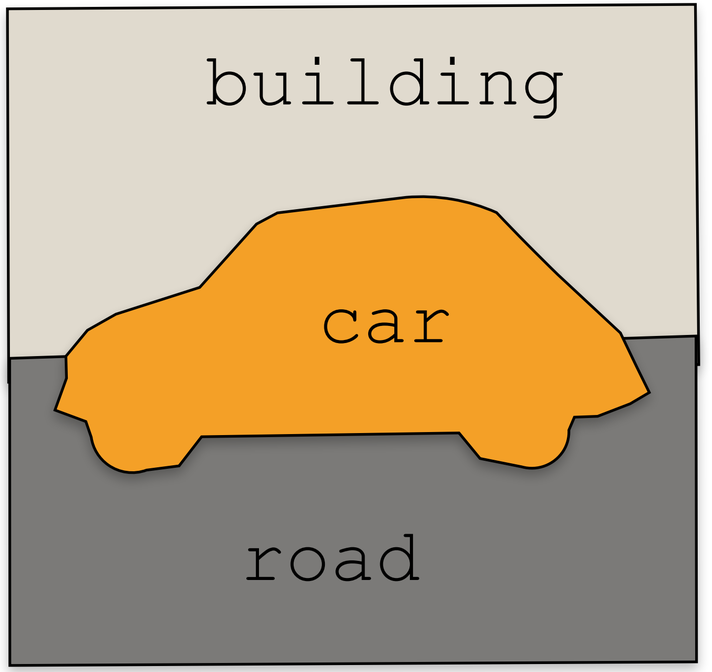
Pre-trained
Classification
Backbone







Segmentation
Head



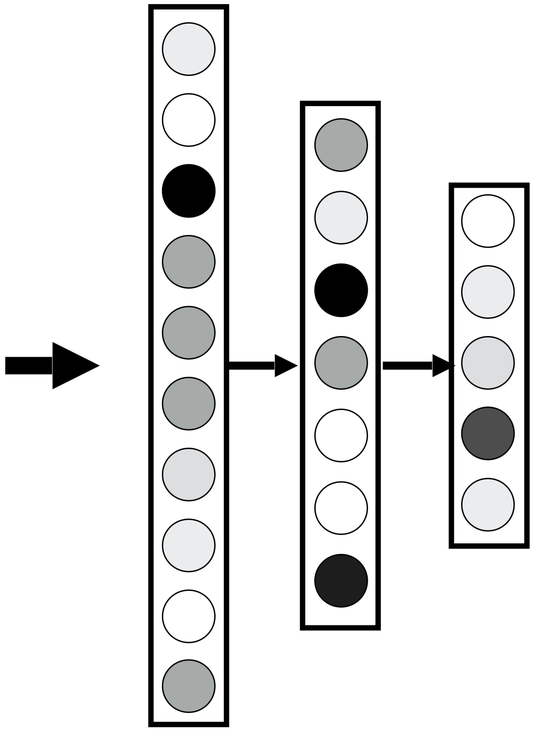
Benefits: Better performance from less data, less compute, fast…

Conclusion: Representation Learning = Pre-training a model for later fine-tuning

$\mathbf{x}$



Image



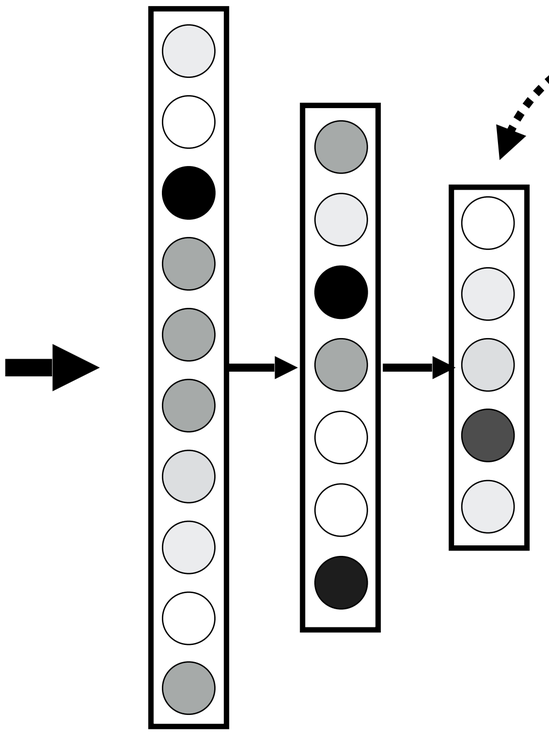
Given a signal, extract a *latent representation* of it that is useful to downstream tasks.

Conclusion: Representation Learning = Pre-training a model for later fine-tuning

$\mathbf{x}$



Image



layer 3 representation of image

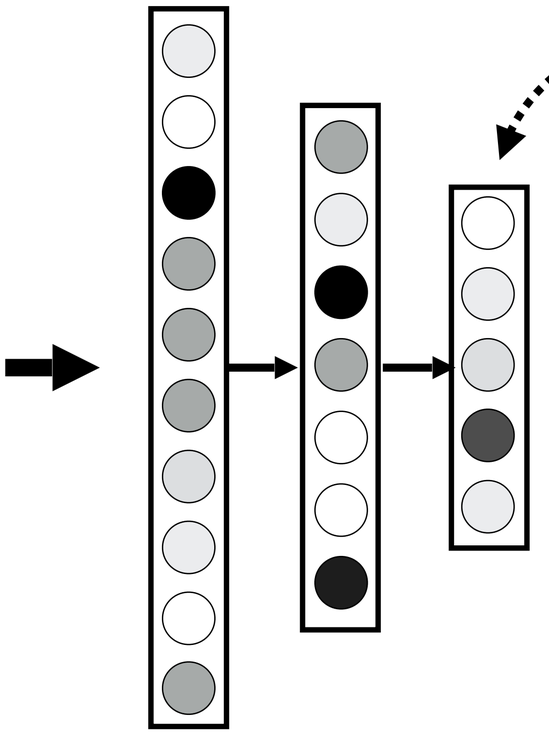
Given a signal, extract a *latent representation* of it that is useful to downstream tasks.

Conclusion: Representation Learning = Pre-training a model for later fine-tuning

$\mathbf{x}$



Image



layer 3 representation of image

layer 1 representation of image

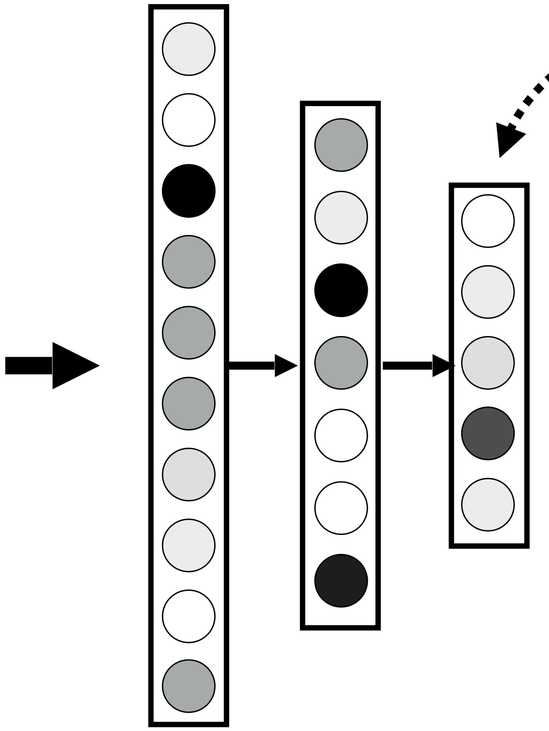
Given a signal, extract a *latent representation* of it that is useful to downstream tasks.

Conclusion: Representation Learning = Pre-training a model for later fine-tuning

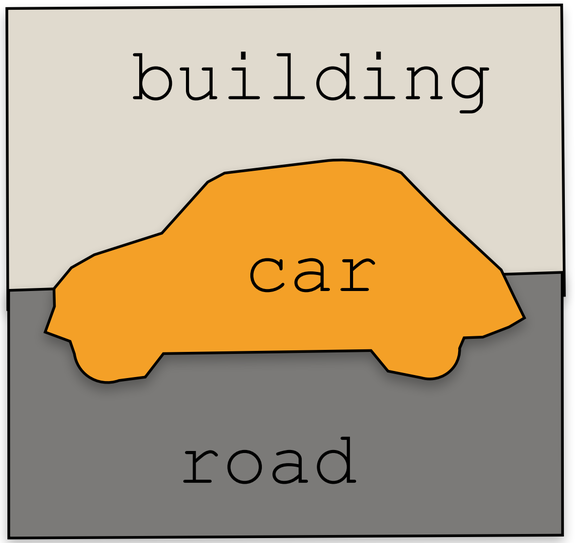
$\mathbf{x}$



Image



layer 3 representation of image



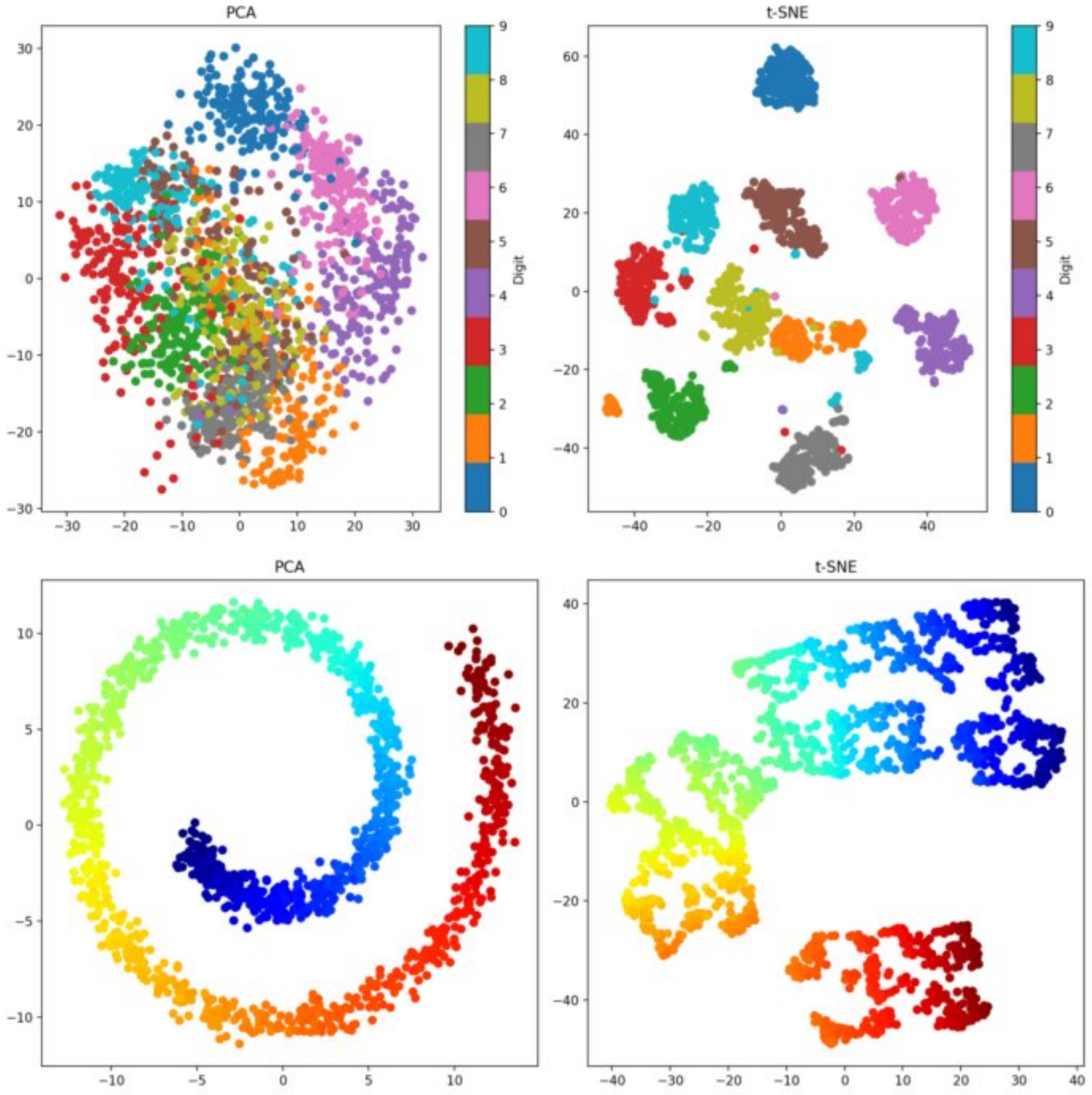
layer 1 representation of image

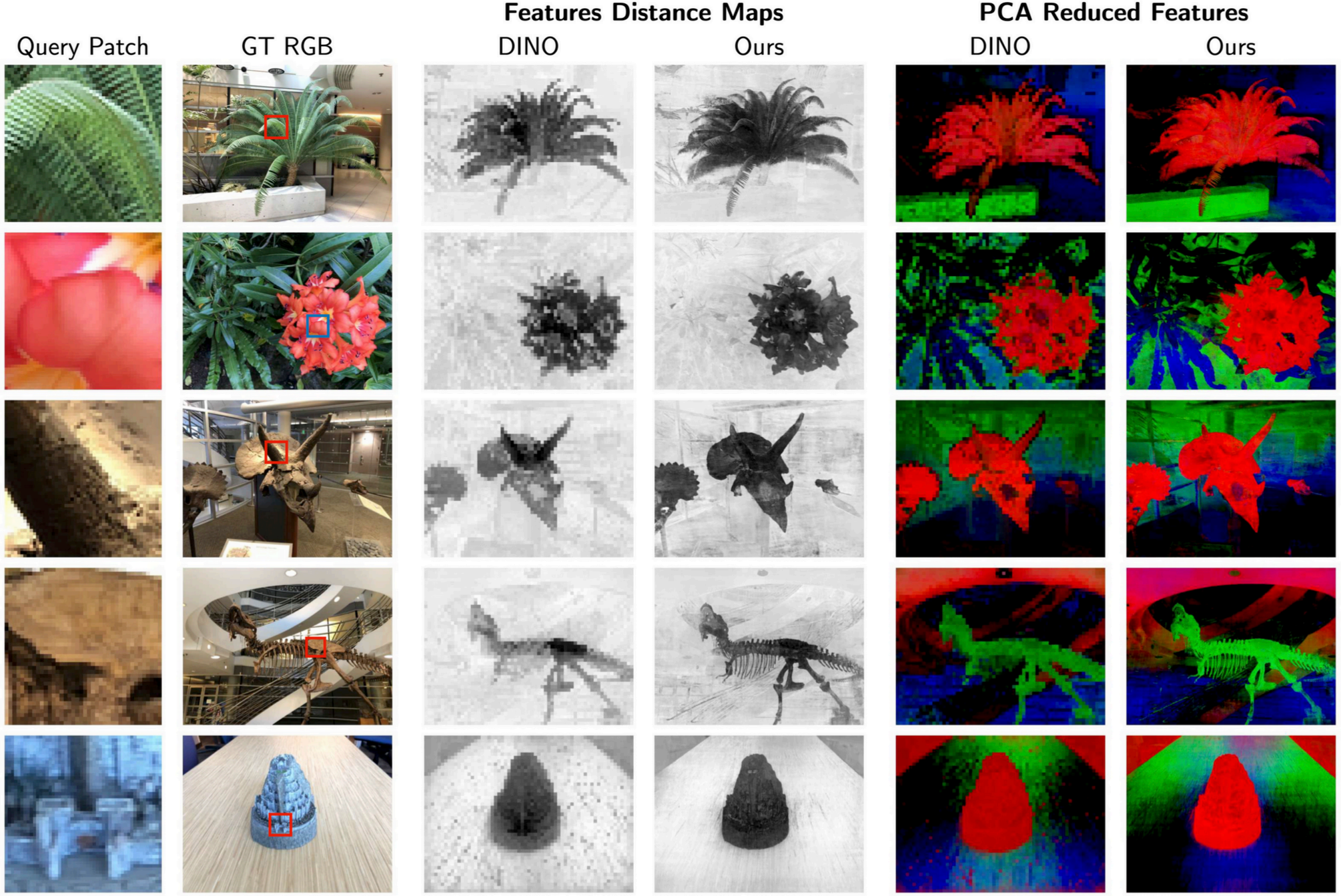
Given a signal, extract a *latent representation* of it that is useful to downstream tasks.

How do we test if a representation is “good”?

- Dimensionality Reduction & Visualization
- Check downstream task performance of features
  - Train small “linear probe” on top of representation
  - Fine-tune network for some downstream task

https://www.fabriziomusacchio.com/blog/2023-06-12-tsne_vs_pca/





From Neural Feature Fusion Fields: 3D Distillation of Self-Supervised 2D Image Representations

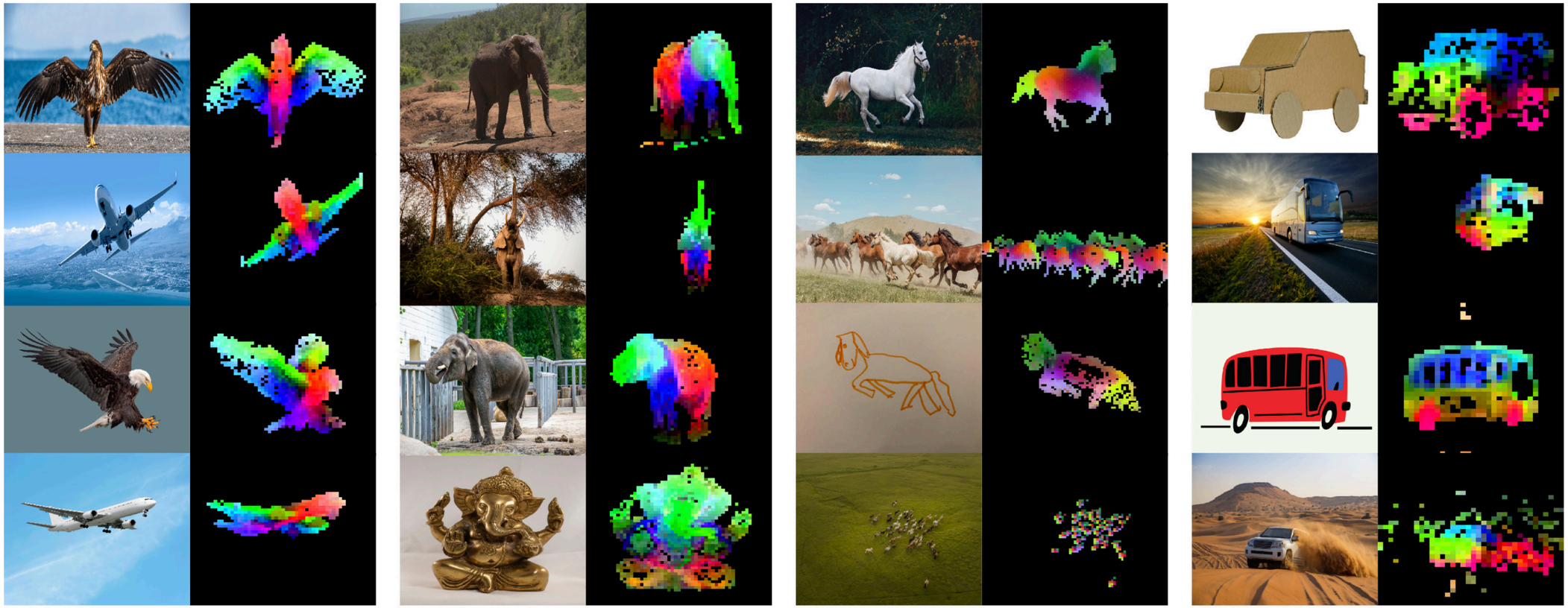


Figure 1: **Visualization of the first PCA components.** We compute a PCA between the patches of the images from the same column (a, b, c and d) and show their first 3 components. Each component is matched to a different color channel. Same parts are matched between related images despite changes of pose, style or even objects. Background is removed by thresholding the first PCA component.

From DinoV2: Learning Robust Visual Features without Supervision, Oquab et al. 2024

Table 2: **Linear and k-NN classification on ImageNet.** We report top-1 accuracy for linear and k-NN evaluations on the validation set of ImageNet for different self-supervised methods. We focus on ResNet-50 and ViT-small architectures, but also report the best results obtained across architectures. * are run by us. We run the k-NN evaluation for models with official released weights. The throughput (im/s) is calculated on a NVIDIA V100 GPU with 128 samples per forward. Parameters (M) are of the feature extractor.

Method	Arch.	Param.	im/s	Linear	k-NN
Supervised	RN50	23	1237	79.3	79.3
SCLR [11]	RN50	23	1237	69.1	60.7
MoCov2 [13]	RN50	23	1237	71.1	61.9
InfoMin [54]	RN50	23	1237	73.0	65.3
BarlowT [66]	RN50	23	1237	73.2	66.0
OBoW [21]	RN50	23	1237	73.8	61.9
BYOL [23]	RN50	23	1237	74.4	64.8
DCv2 [9]	RN50	23	1237	75.2	67.1
SwAV [9]	RN50	23	1237	**75.3**	65.7
DINO	RN50	23	1237	**75.3**	**67.5**
Supervised	ViT-S	21	1007	79.8	79.8
BYOL* [23]	ViT-S	21	1007	71.4	66.6
MoCov2* [13]	ViT-S	21	1007	72.7	64.4
SwAV* [9]	ViT-S	21	1007	73.5	66.3
DINO	ViT-S	21	1007	**77.0**	**74.5**

From DINO (Emerging Properties in Self-Supervised Vision Transformers), Caron et al. 2021

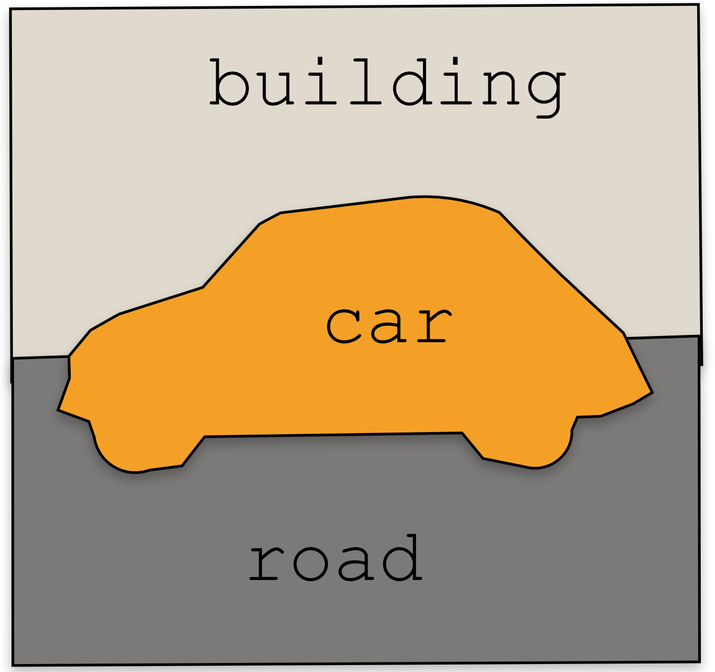
		NYUd (0.330)			KITTI (2.10)			NYUd → SUN RGB-D (0.421)		
Method	Arch.	lin. 1	lin. 4	DPT	lin. 1	lin. 4	DPT	lin. 1	lin. 4	DPT
OpenCLIP	ViT-G/14	0.541	0.510	0.414	3.57	3.21	2.56	0.537	0.476	0.408
MAE	ViT-H/14	0.517	0.483	0.415	3.66	3.26	2.59	0.545	0.523	0.506
DINO	ViT-B/8	0.555	0.539	0.492	3.81	3.56	2.74	0.553	0.541	0.520
iBOT	ViT-L/16	0.417	0.387	0.358	3.31	3.07	2.55	0.447	0.435	0.426
DINOv2	ViT-S/14	0.449	0.417	0.356	3.10	2.86	2.34	0.477	0.431	0.409
	ViT-B/14	0.399	0.362	0.317	2.90	2.59	2.23	0.448	0.400	0.377
	ViT-L/14	0.384	0.333	0.293	2.78	2.50	2.14	0.429	0.396	0.360
	ViT-g/14	**0.344**	**0.298**	**0.279**	**2.62**	**2.35**	**2.11**	**0.402**	**0.362**	**0.338**

Table 11: **Depth estimation with frozen features**. We report performance when training a linear classifier on top of one (lin. 1) or four (lin. 4) transformer layers, as well, as the DPT decoder (DPT) of Ranftl et al. (2021). We report the RMSE metric on the 3 datasets. Lower is better. For reference, we report state-of-the-art results taken from Li et al. (2022b) on each benchmark on top of the Table.

From DinoV2: Learning Robust Visual Features without Supervision, Oquab et al. 2024

Can we find a pre-training objective *without* labels?

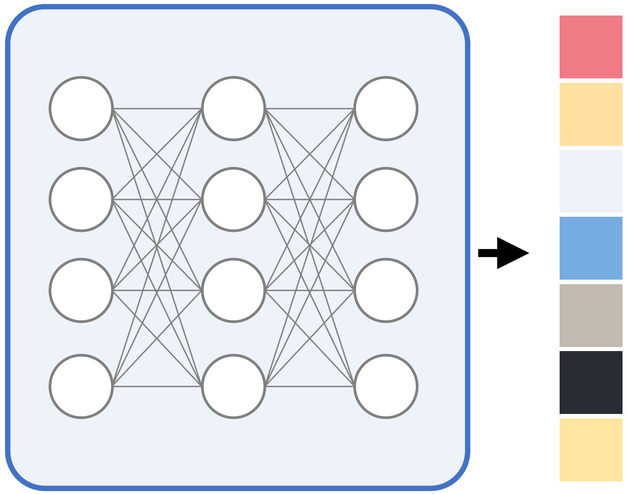


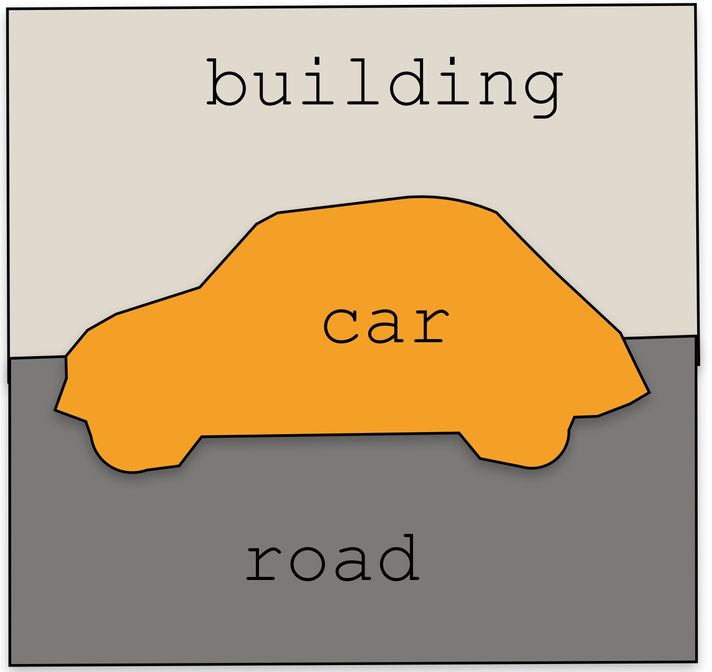


Can we find a pre-training objective *without* labels?



Pre-trained
Classification
Backbone

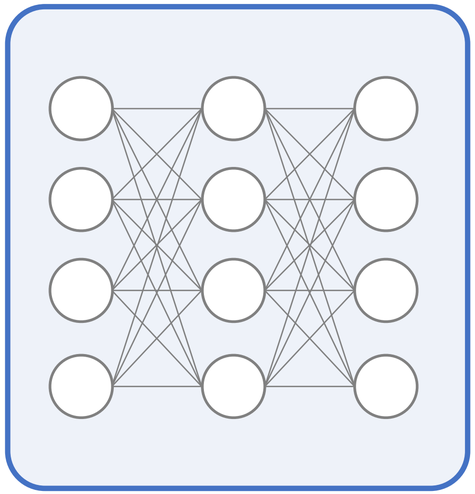




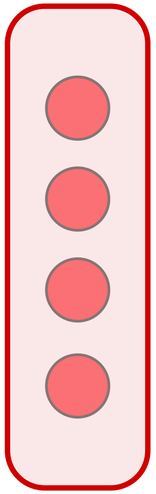
Can we find a pre-training objective *without* labels?



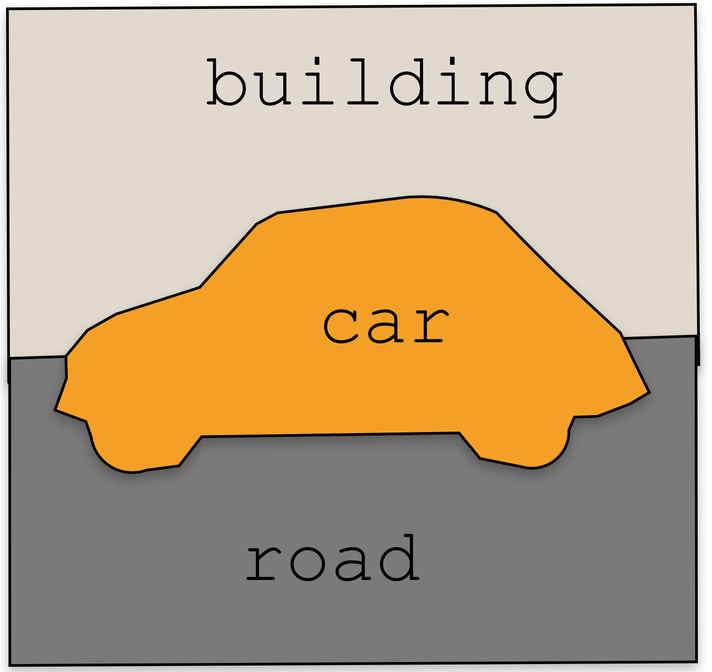
Pre-trained
Classification
Backbone







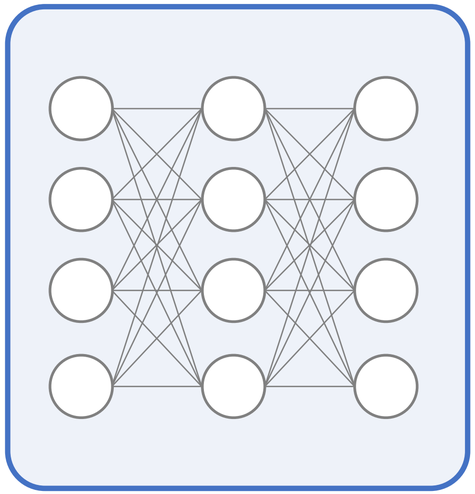
Segmentation
Head



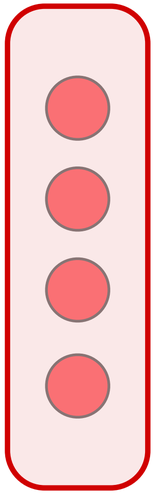
Can we find a pre-training objective *without* labels?



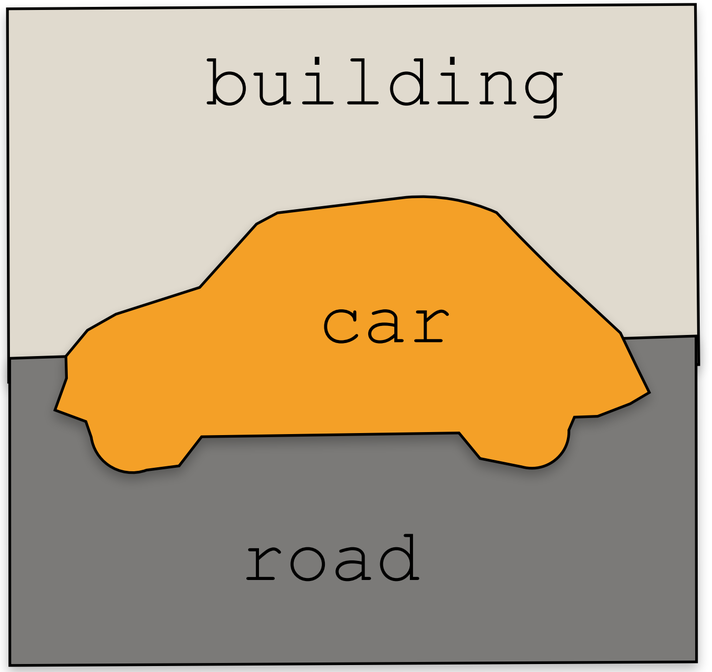
Pre-trained
Classification
Backbone





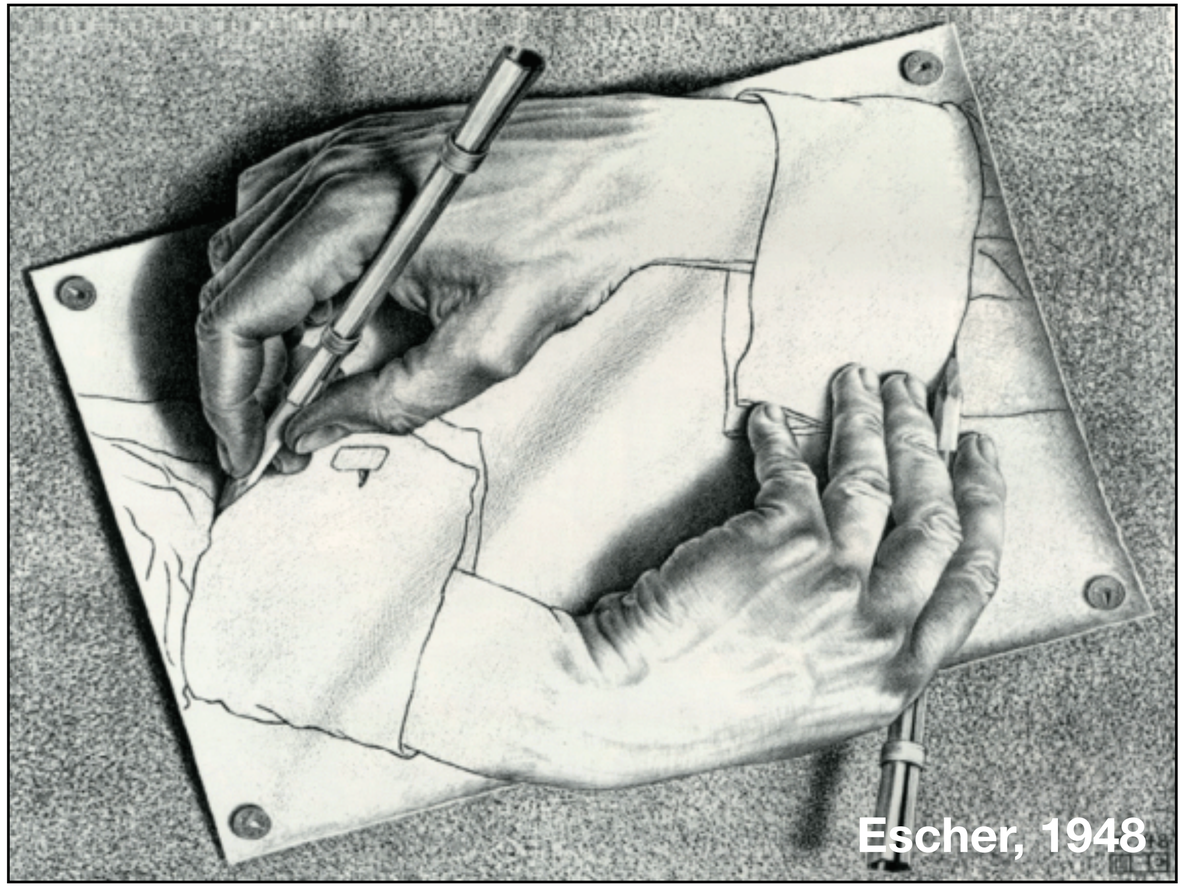


Segmentation
Head

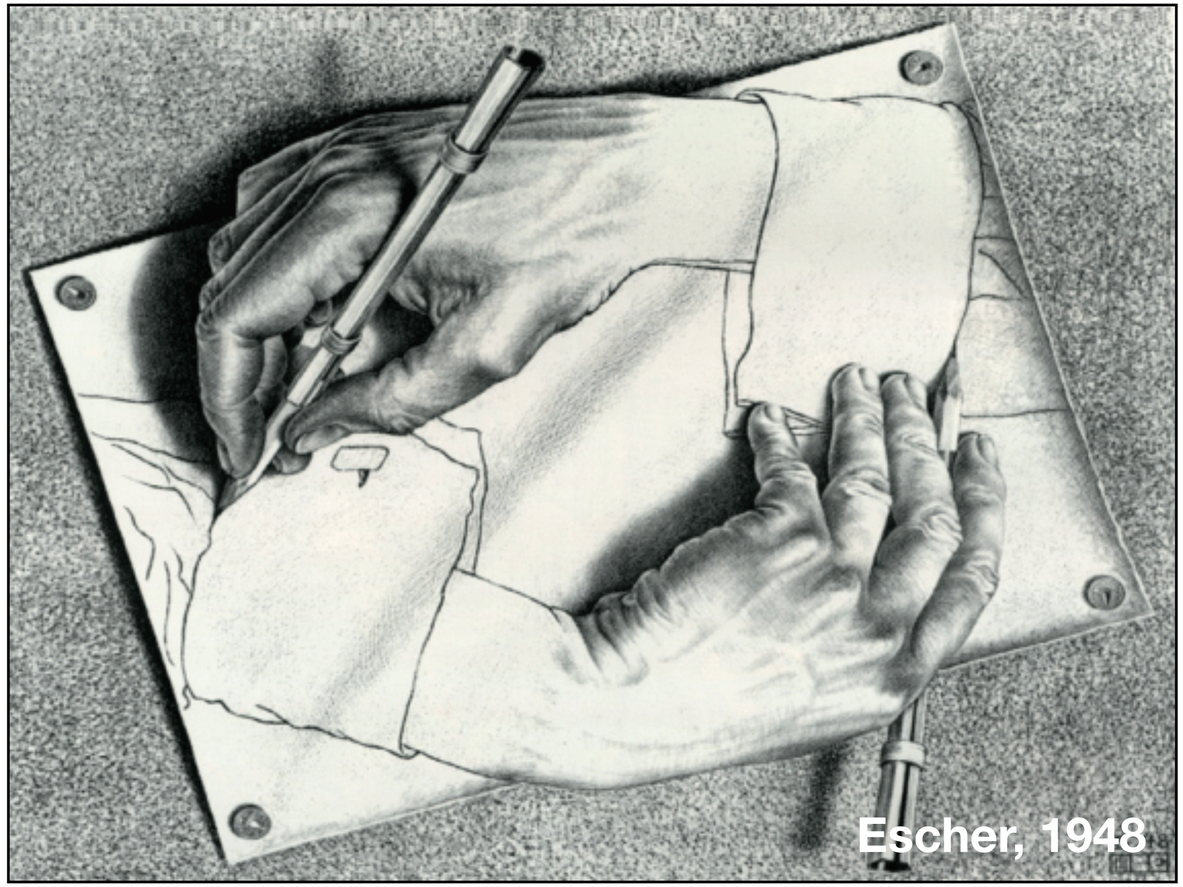


Benefits: Better performance from less data, less compute, fast…

Self-supervised learning

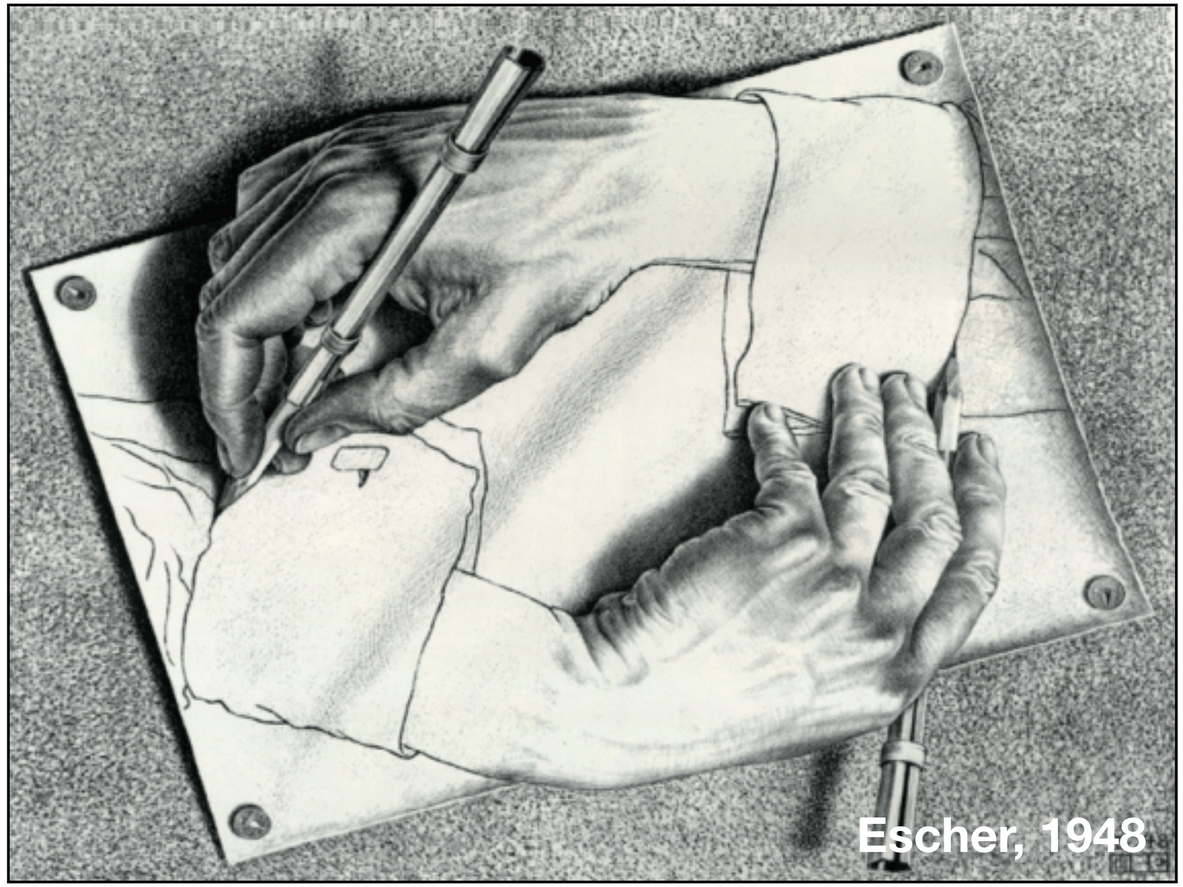


Self-supervised learning



Common trick:

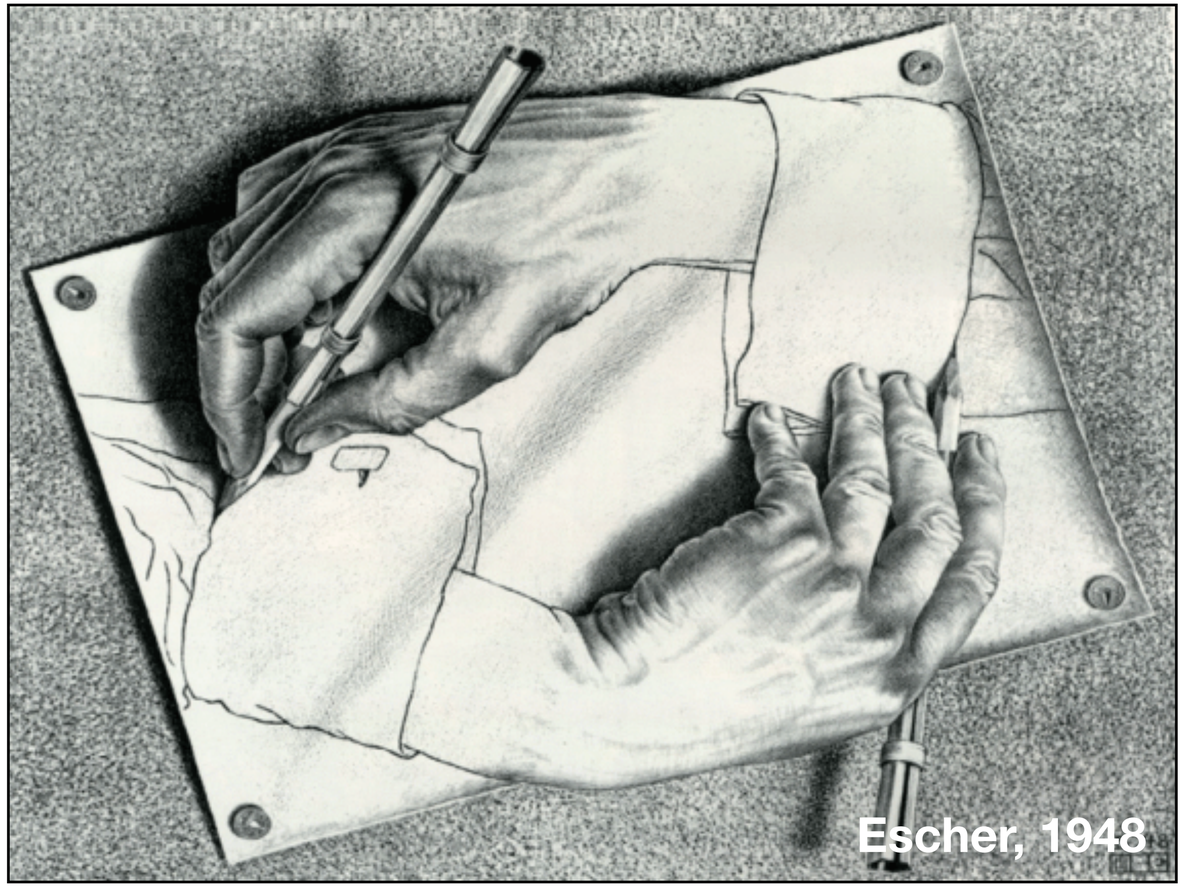
Self-supervised learning



Common trick:

- Convert “unsupervised” problem into “supervised” empirical risk minimization

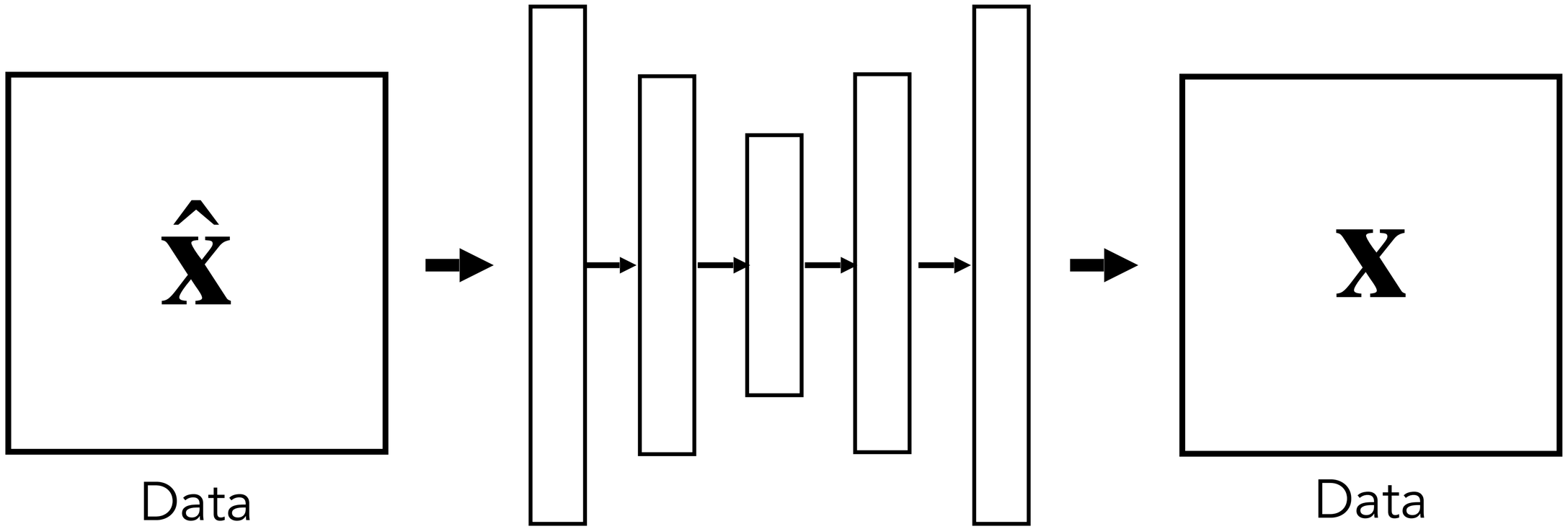
Self-supervised learning



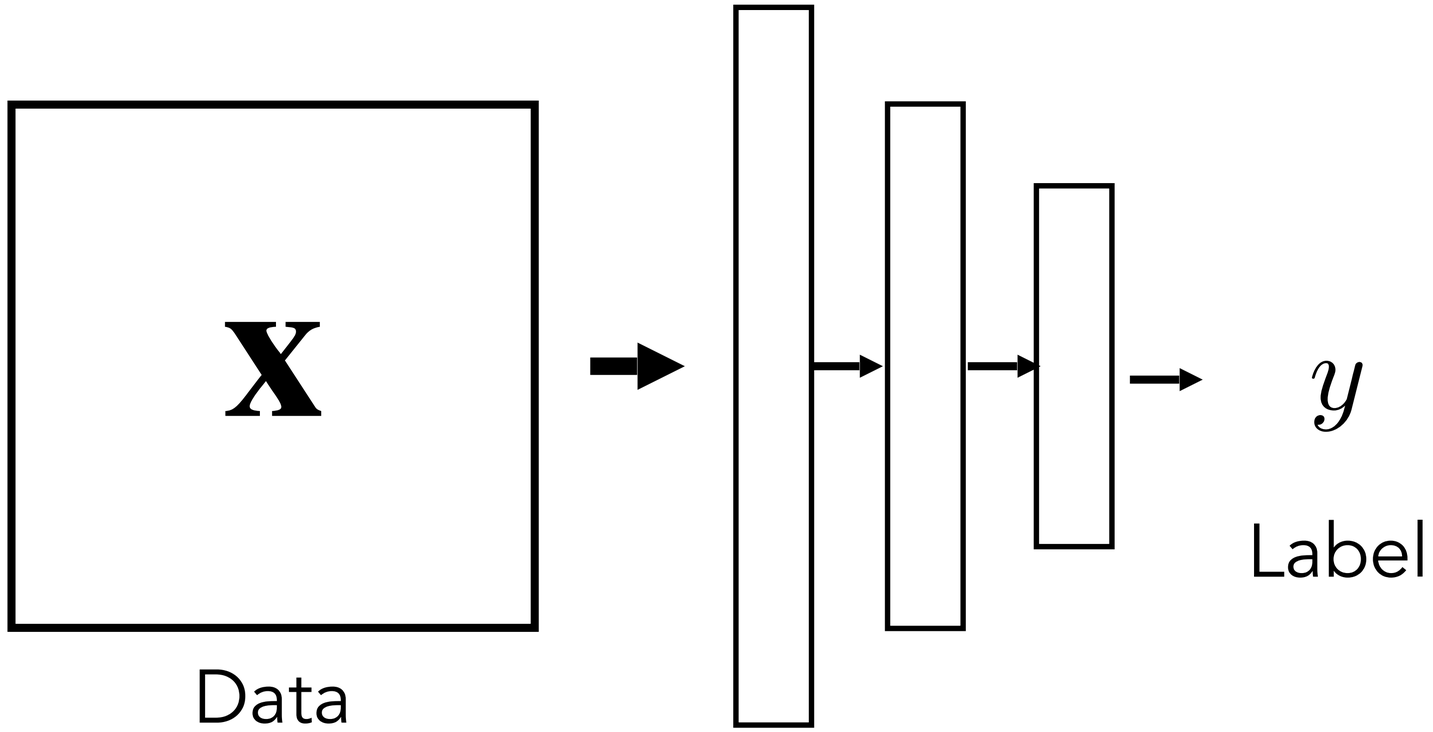
Common trick:

- Convert “unsupervised” problem into “supervised” empirical risk minimization
- Do so by cooking up “labels” (prediction targets) from the raw data itself — called **pretext task**

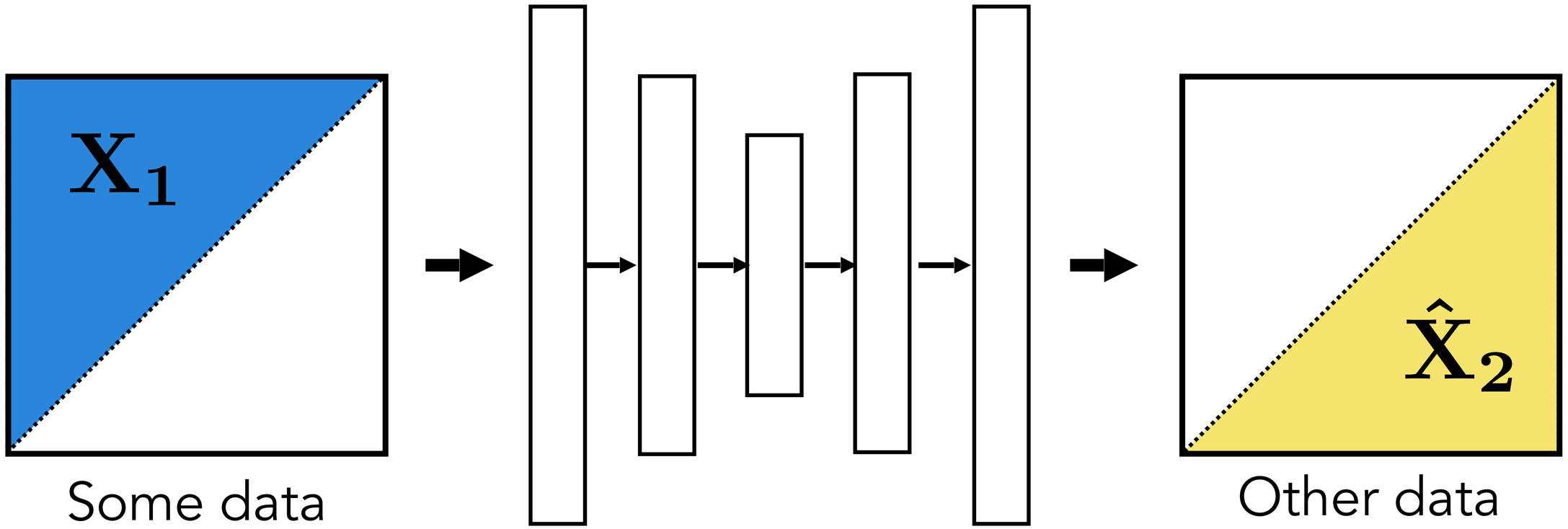
Data compression

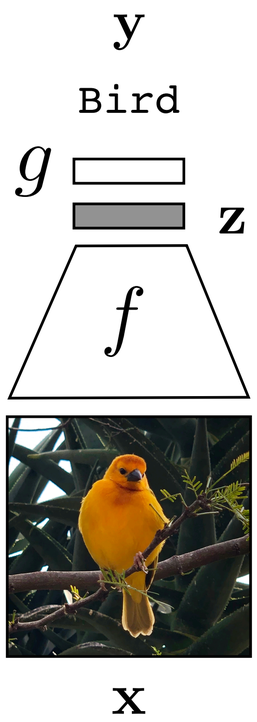
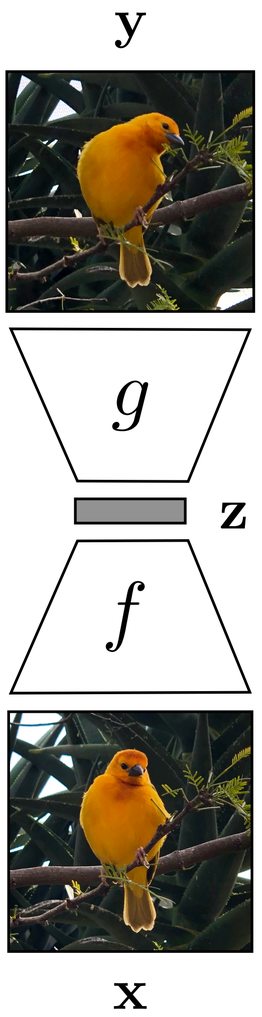
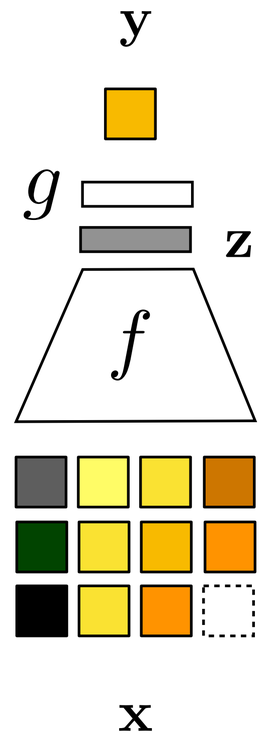


Label prediction

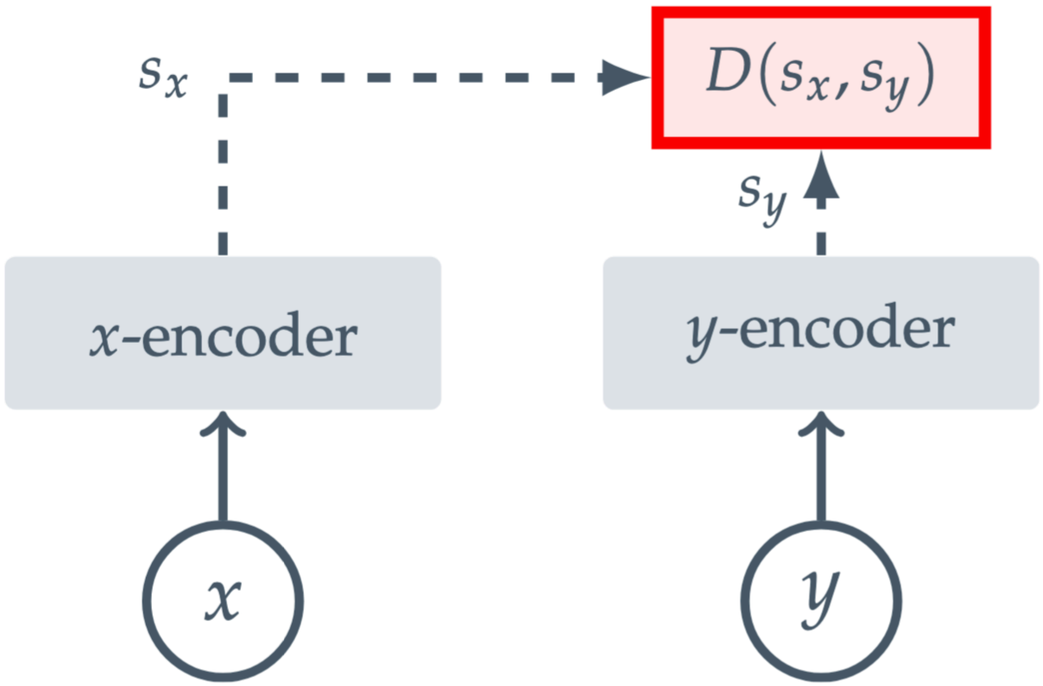


Data prediction
aka "self-supervised learning"



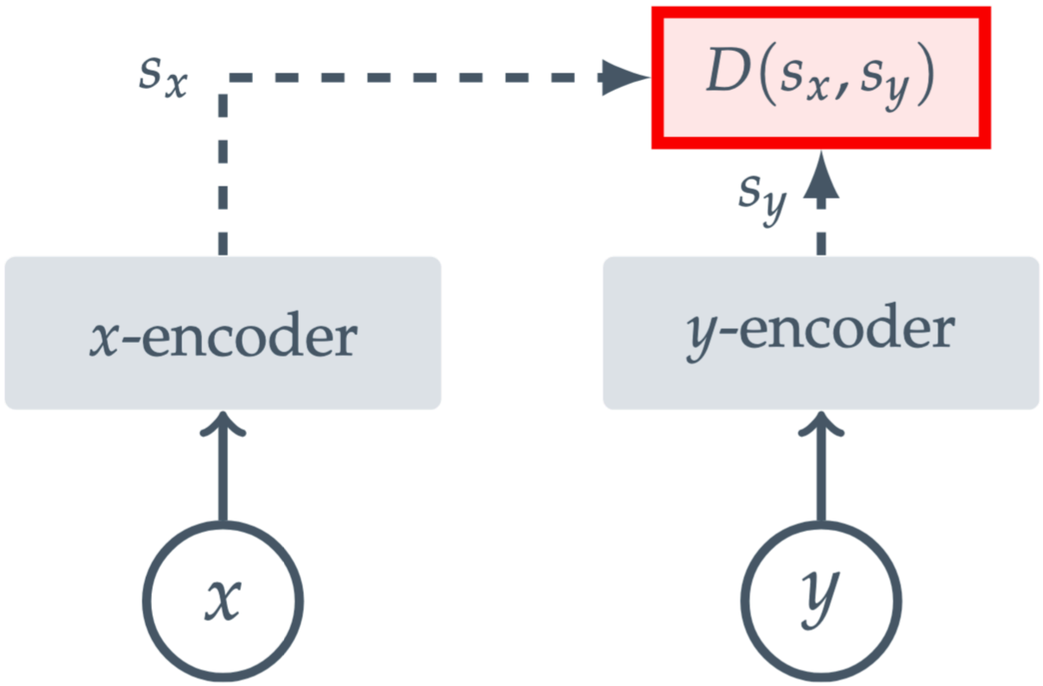
Pretext task:	Class prediction	Future frame prediction	Next pixel prediction
Model schematic:			

Joint Embedding Architectures



Given *compatible* x, y pairs (i.e. things that belong together, such as augmentations of an image, two similar images, etc), learn encoders that output s_x, s_y that are similar according to some distance metric D.

Problem: Collapse / Trivial Solution

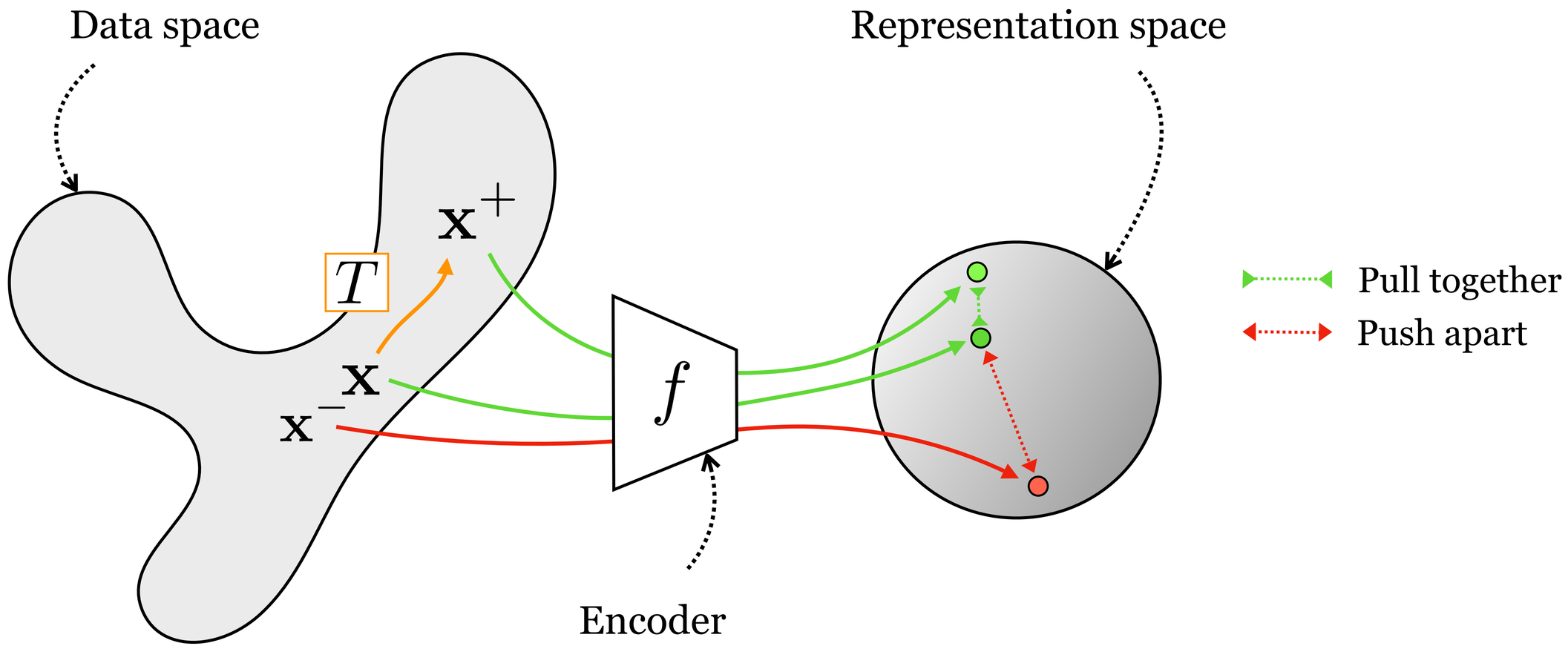


Trivial solution to

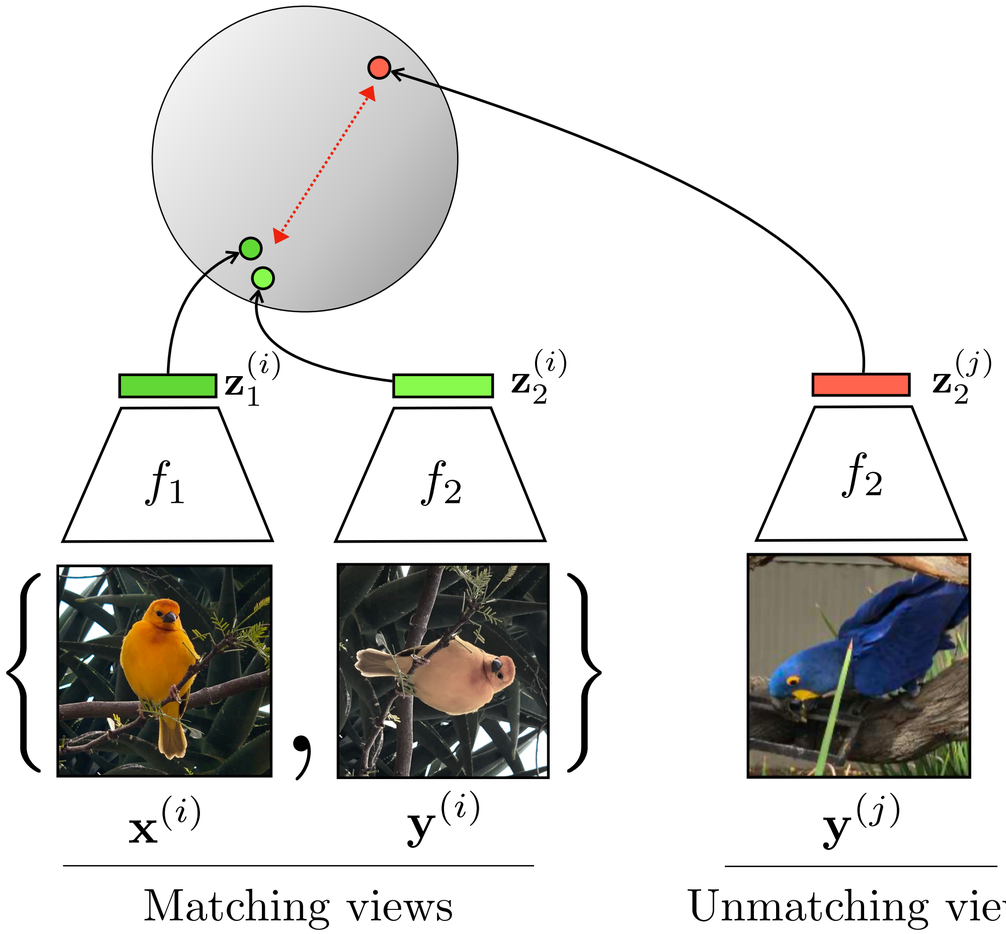
$$\min_{s_x, s_y} D(s_x, s_y)$$

Is always $s_x = s_y = c$ for some constant c (such as 0).

Contrastive Learning



Contrastive Learning — Transformations



Data

$\{\mathbf{x}^{(i)}\}_{i=1}^{N}, T \longrightarrow$

Constrastive learning (transformations)

Objective

$\sum_{i,j} D(f(T(\mathbf{x}^{(i)})), f(\mathbf{x}^{(i)}))$

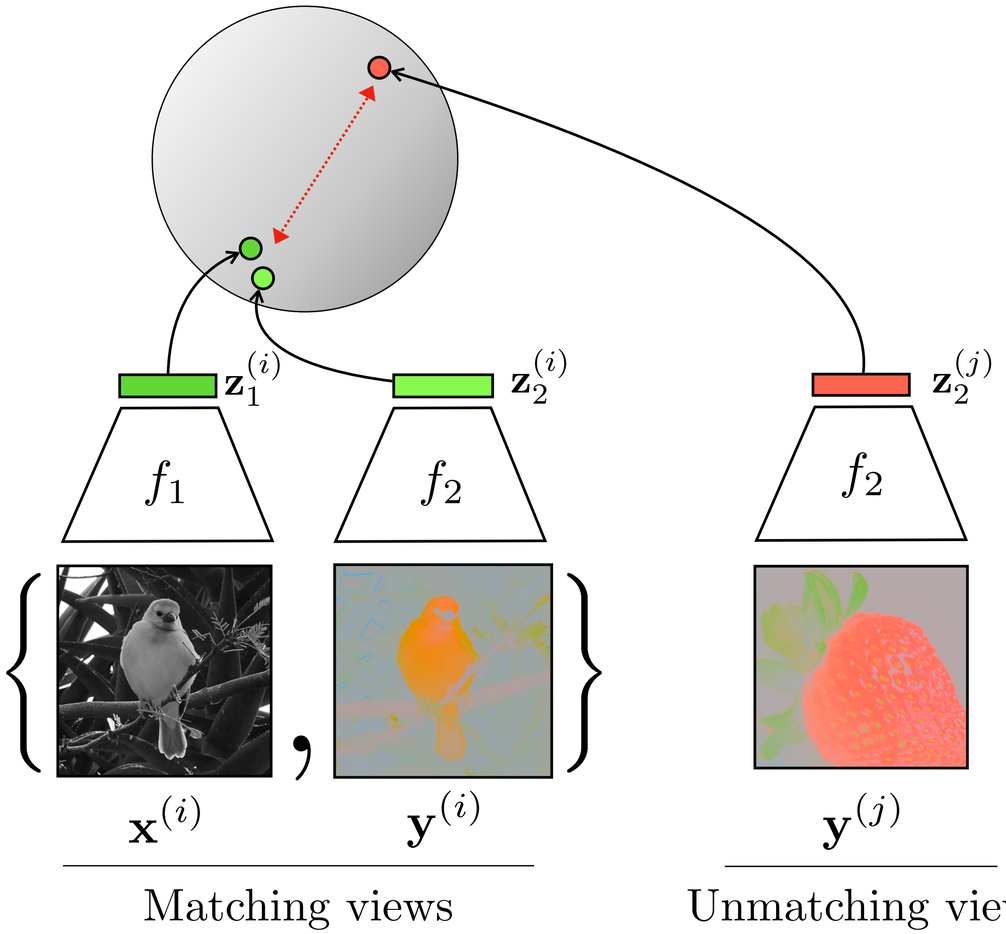
$-D(f(\mathbf{x}^{(i)}), f(\mathbf{x}^{(j)}))$

Hypothesis space

$f : \mathbb{R}^N \rightarrow \mathbb{R}^M$

$\longrightarrow f$

Contrastive Learning — Co-occurrences



Data

$\{\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\}_{i=1}^{N} \rightarrow$

Constrastive learning (co-occurrences)

Objective

$$\sum_{i,j} D(f_1(\mathbf{x}^{(i)}), f_2(\mathbf{y}^{(i)}))$$

$$-D(f_1(\mathbf{x}^{(i)}), f_2(\mathbf{y}^{(j)}))$$

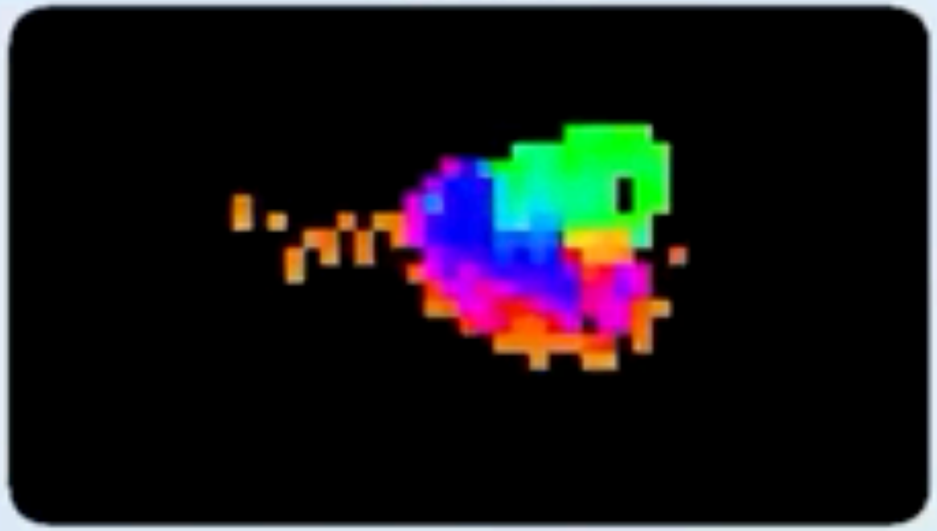
Hypothesis space

$$f_1, f_2 : \mathbb{R}^N \rightarrow \mathbb{R}^M$$

$\rightarrow f_1, f_2$



DINO



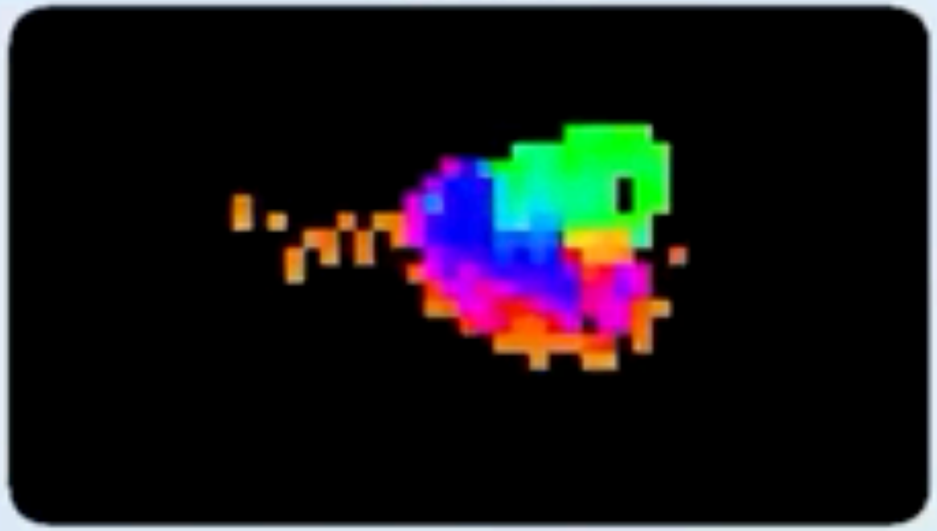
DINOv2



DINOv2, Oquab et al., arXiv 2023



DINO



DINOv2



DINOv2, Oquab et al., arXiv 2023

The Model Zoo of Joint Embedding Architectures

- DINO
- BYOL
- VIC-Reg
- Info-NCE
- LeJEPA
- I-JEPA
- …

SSL pre-training dominates any and all “conventional” vision benchmarks (3D reconstruction, classification, segmentation, detection…)

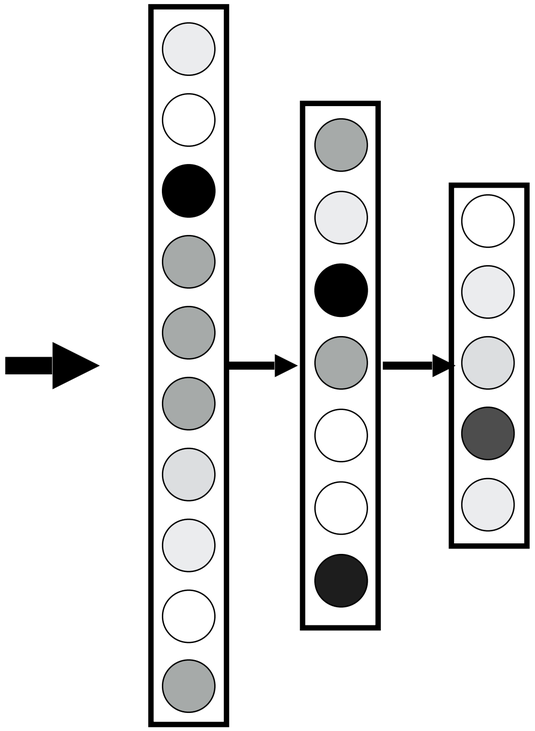
We will talk about some of these in the lecture!

Conclusion: Representation Learning = Pre-training a model for later fine-tuning

$\mathbf{x}$



Image



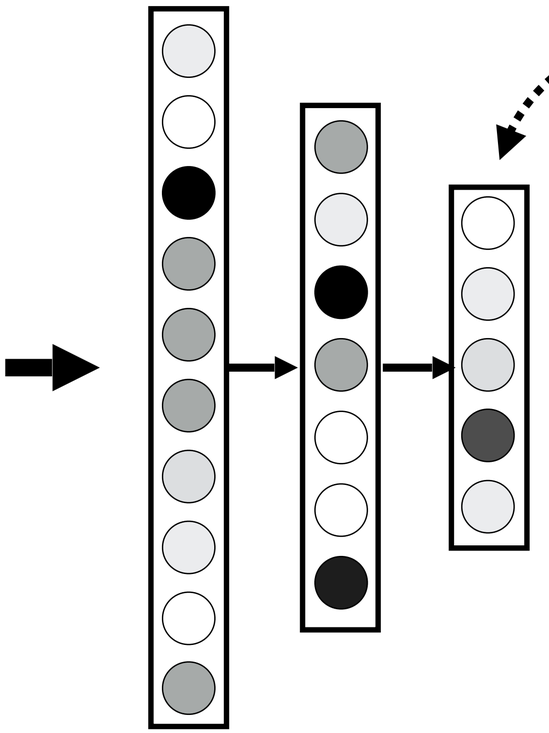
Given a signal, extract a *latent representation* of it that is useful to downstream tasks.

Conclusion: Representation Learning = Pre-training a model for later fine-tuning

$\mathbf{x}$



Image



layer 3 representation of image

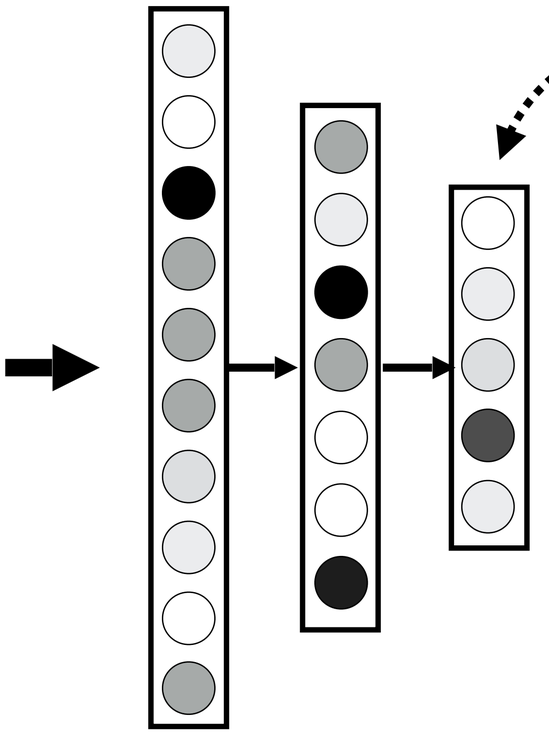
Given a signal, extract a *latent representation* of it that is useful to downstream tasks.

Conclusion: Representation Learning = Pre-training a model for later fine-tuning

$\mathbf{x}$



Image



layer 3 representation of image

layer 1 representation of image

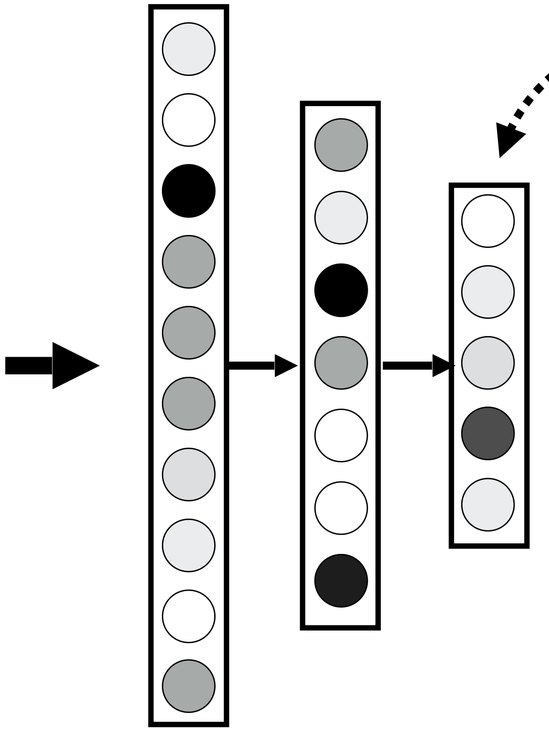
Given a signal, extract a *latent representation* of it that is useful to downstream tasks.

Conclusion: Representation Learning = Pre-training a model for later fine-tuning

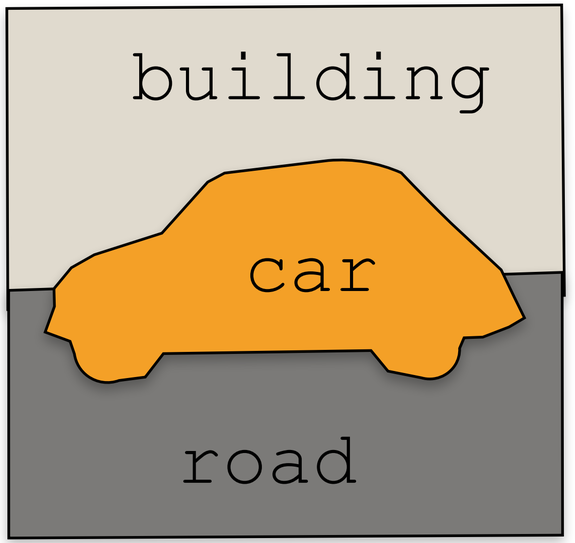
$\mathbf{x}$



Image



layer 3 representation of image



layer 1 representation of image

Given a signal, extract a *latent representation* of it that is useful to downstream tasks.

What is generative modeling?

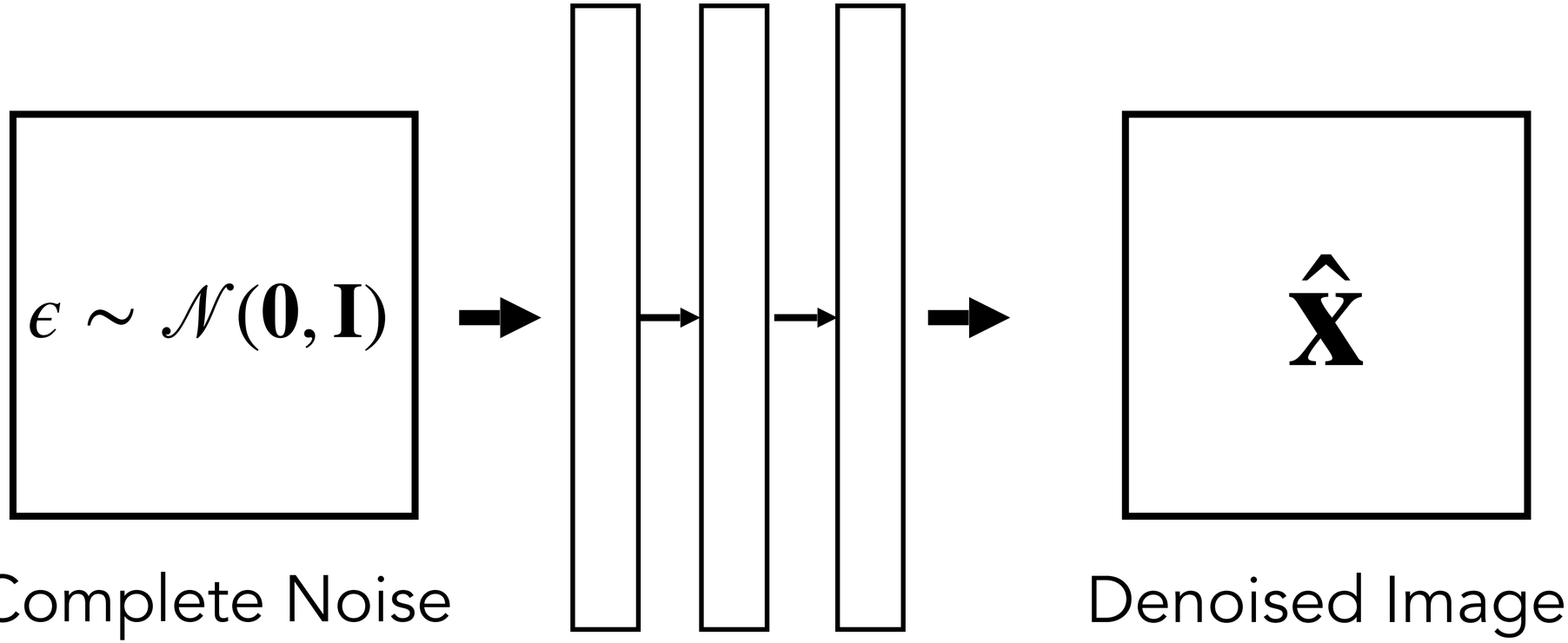
> *Creating noise from data is easy;*
> *creating data from noise is generative modeling*
>
> - Song et al. 2020

What is generative modeling?

> *Creating noise from data is easy;*
> *creating data from noise is generative modeling*
>
> - Song et al. 2020

- Given samples from some distribution,
  learn to generate new samples
- In practice: learn to map noise to (images, audio, …)

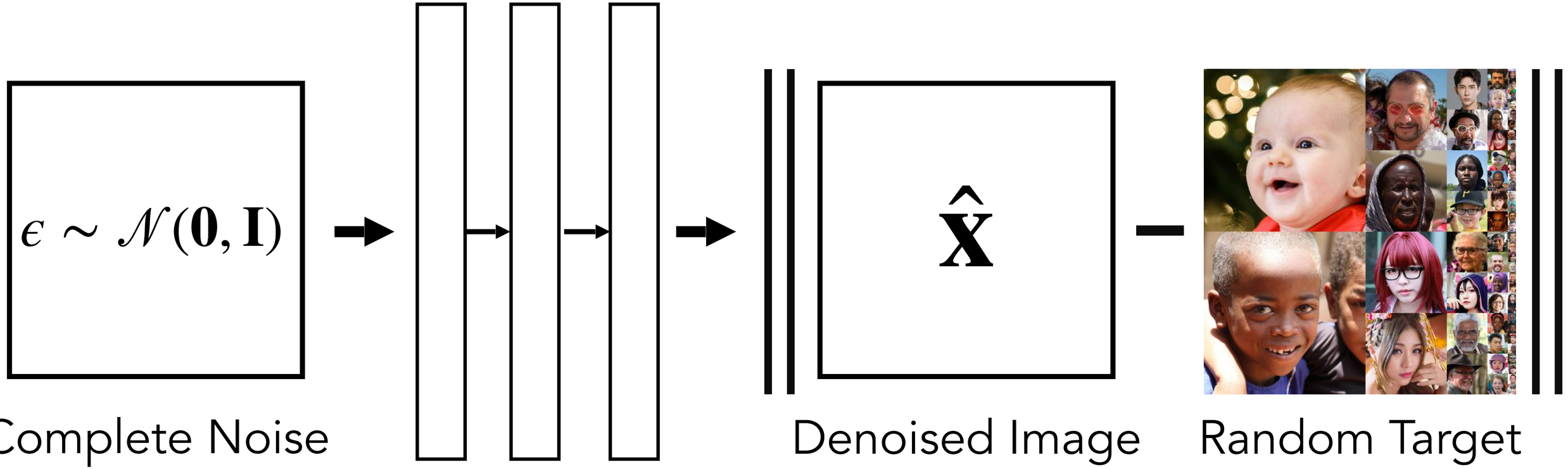
A naive but valuable perspective on GenAI



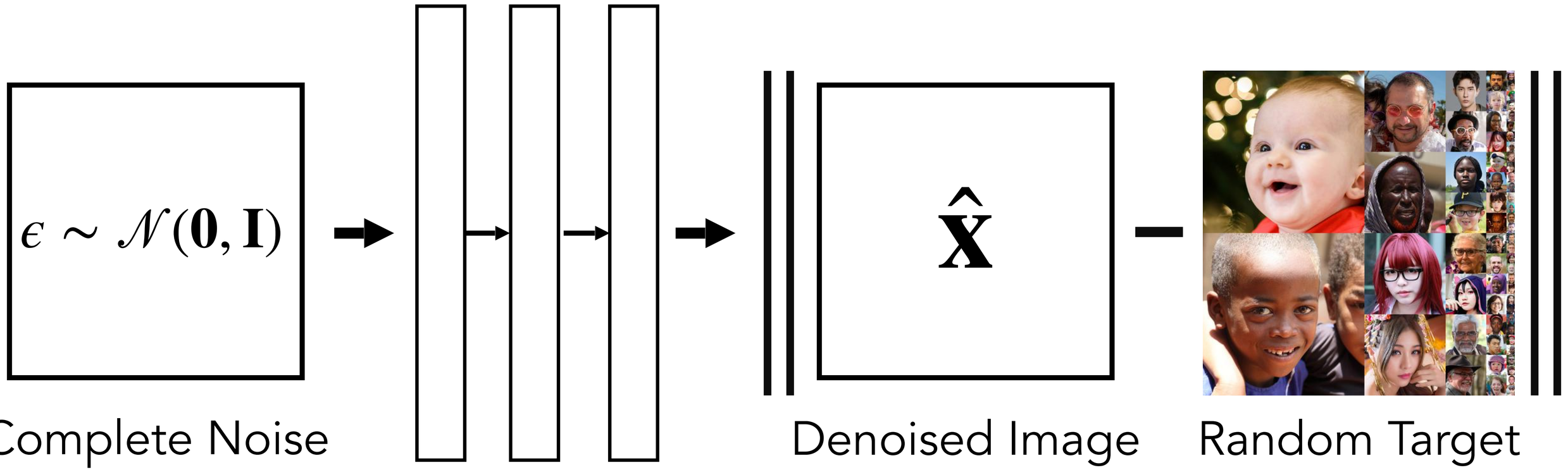
Complete Noise

Denoised Image

A naive but valuable perspective on GenAI



A naive but valuable perspective on GenAI

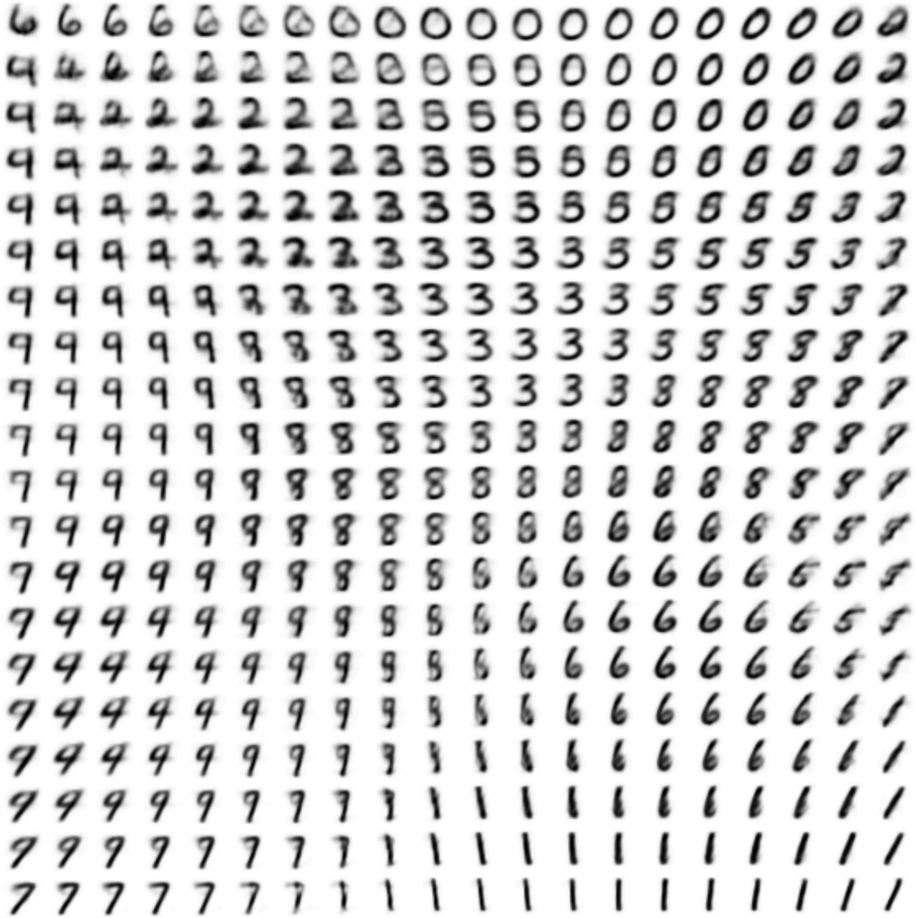


What will happen?

2013: VAEs



(a) Learned Frey Face manifold

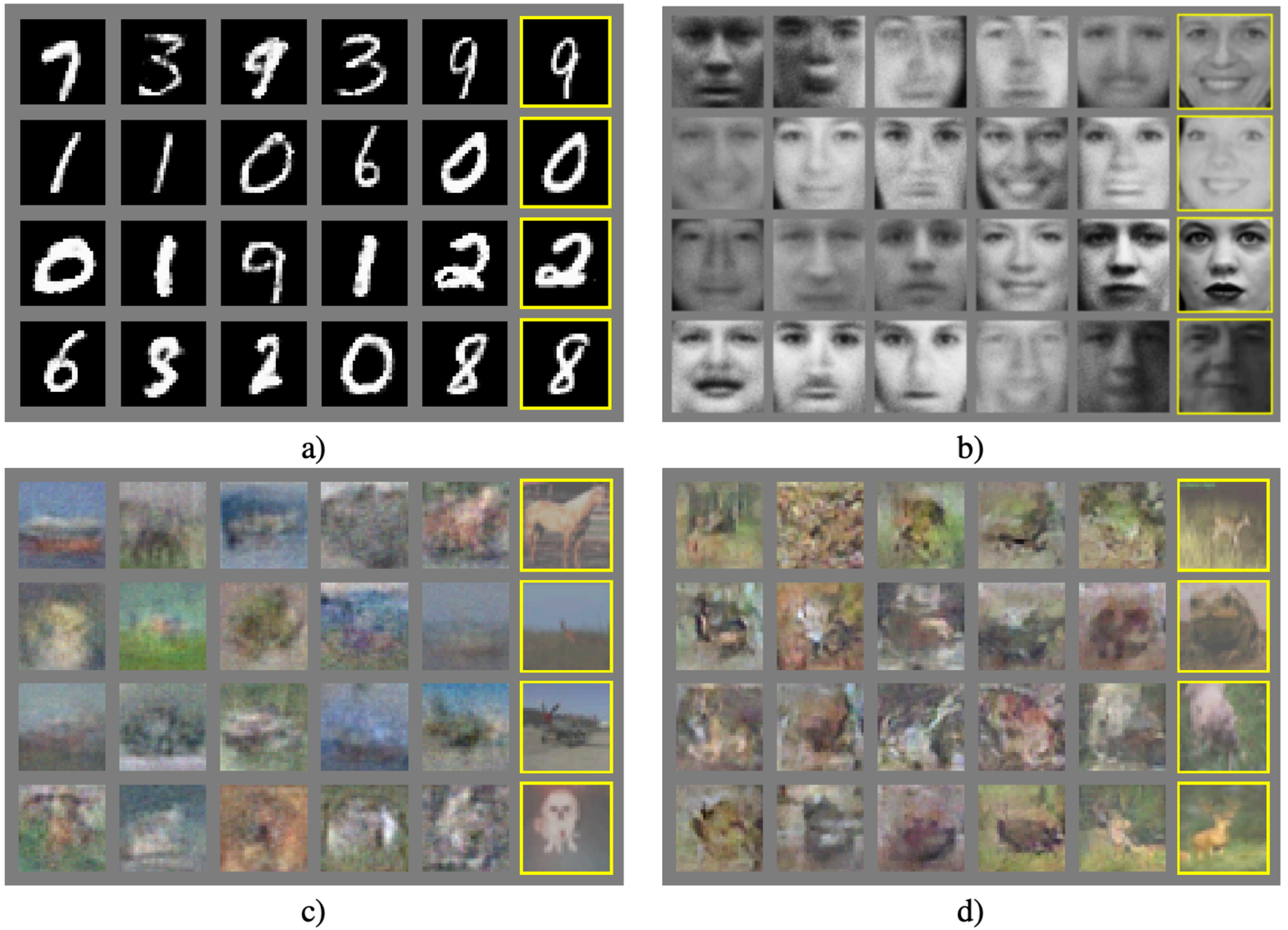


(b) Learned MNIST manifold

Video source: https://x.com/runwayml/status/1807822396415467686

Auto-encoding Variational Bayes, Kingma et al. 2013

2014: GANs



Generative Adversarial Networks, Goodfellow et al. 2014

GANs by 2019



Figure 1: Class-conditional samples generated by our model.

LARGE SCALE GAN TRAINING FOR HIGH FIDELITY NATURAL IMAGE SYNTHESIS, Brock et al. 2019

VAEs by 2020



Figure 1: 256×256-pixel samples generated by NVAE, trained on CelebA HQ [28].

NVAE: A Deep Hierarchical Variational Autoencoder, Vahdat et al. 2020

2020: Diffusion

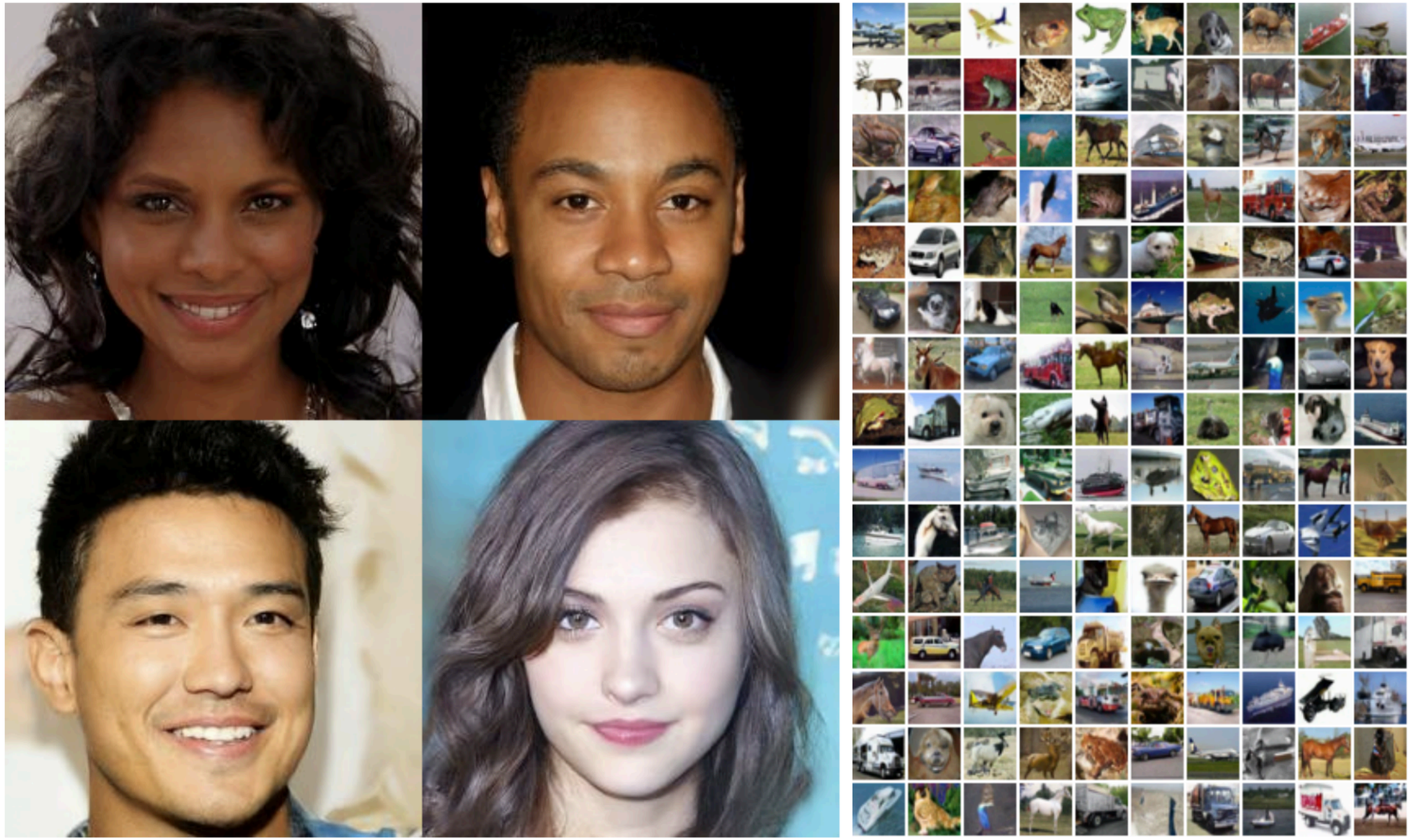


Figure 1: Generated samples on CelebA-HQ 256 × 256 (left) and unconditional CIFAR10 (right)

Deep Unsupervised Learning using Nonequilibrium Thermodynamics, Sohl-Dickstein et al. 2015
Denoising Diffusion Probabilistic Models, Ho et al. 2006

2026: Some New Contenders

Generative Modeling via Drifting

Mingyang Deng [1] He Li [1] Tianhong Li [1] Yilun L

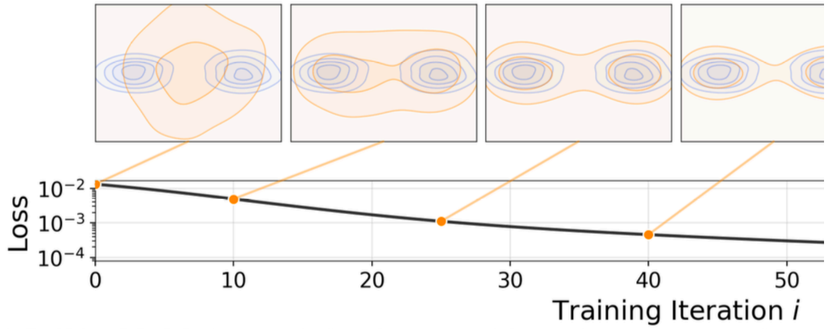


Figure 1. **Drifting Model.** A network f performs a pushforward operation: $q = f_{\#}p_{\text{prio}}$ not shown here) to a pushforward distribution q (orange). The goal of training is to training iterates, we obtain a sequence of models $\{f_i\}$, which corresponds to a sequen Model focuses on the evolution of this pushforward distribution at *training-time*. We i approaches zero when q matches p_{data}. This drifting field provides a loss function (y-a

Image Generation with a Sphere Encoder

Kaiyu Yue [† 1 2] Menglin Jia [† 1] Ji Hou [1] Tom Goldstein [2]

Abstract

We introduce the Sphere Encoder, an efficient generative framework capable of producing images in a single forward pass and competing with many-step diffusion models using fewer than five steps. Our approach works by learning an encoder that maps natural images uniformly onto a spherical latent space, and a decoder that maps random latent vectors back to the image space. Trained solely through image reconstruction losses, the model generates an image by simply decoding a random point on the sphere. Our architecture naturally supports conditional generation, and looping the encoder/decoder a few times can further enhance image quality. Across several datasets, the sphere encoder approach yields performance competitive with state of the art diffusions, but with a small fraction of the inference cost. Project page is available at sphere-encoder.github.io.



Deep Unsupervised Learning using Nonequilibrium Thermodynamics, Sohl-Dickstein et al. 2015
Denoising Diffusion Probabilistic Models, Ho et al. 2006

Lots of unique, new questions in GenAI land…

- Tokenization and latent diffusion: It matters *which space* you do generative modeling in
- Some evidence that representation learners (such as DINO) can help / improve image generation - but why…?
- Can we understand Diffusion Models and other generative models *mechanistically*, i.e. can we understand what they are doing under the hood? Can we model it?

Since 2022: Video Generative Models



Video source: https://x.com/runwayml/status/1807822396415467686

Diffusion Probabilistic Modeling for Video Generation, Yang et al. 2022

Video Diffusion Models, Ho et al. 2022

Since 2022: Video Generative Models

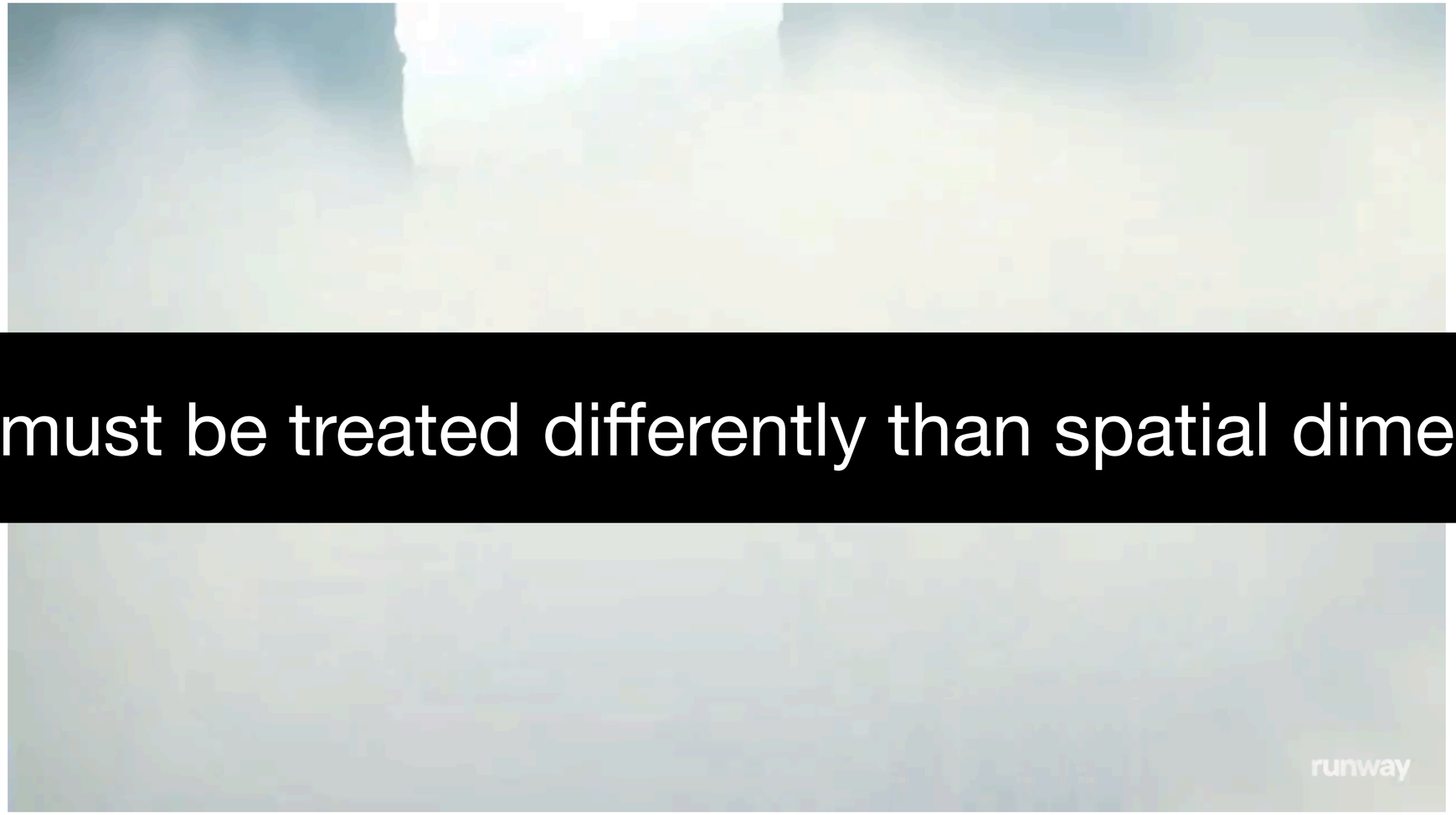


Video source: https://x.com/runwayml/status/1807822396415467686

Diffusion Probabilistic Modeling for Video Generation, Yang et al. 2022

Video Diffusion Models, Ho et al. 2022

Since 2022: Video Generative Models



Video source: https://x.com/runwayml/status/1807822396415467686

Time must be treated differently than spatial dimensions!

Diffusion Probabilistic Modeling for Video Generation, Yang et al. 2022

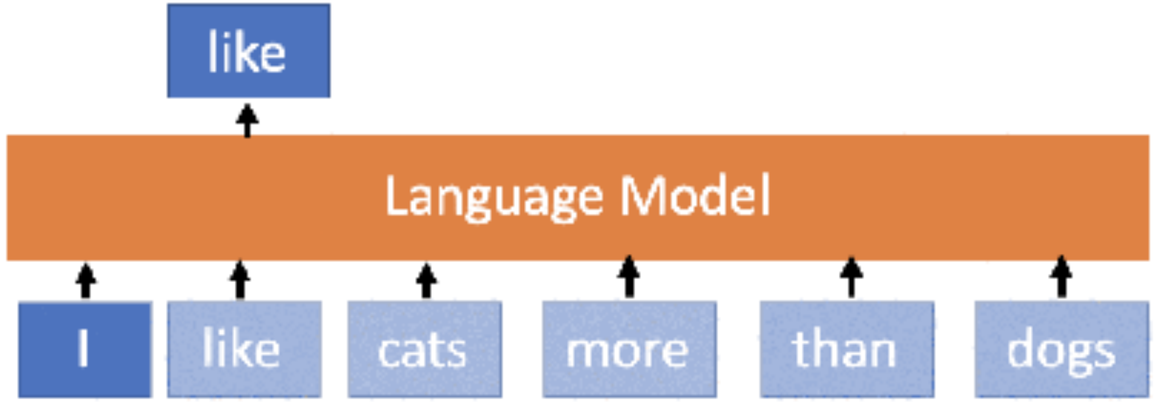
Video Diffusion Models, Ho et al. 2022

Two paradigms of sequence modeling

Full-Sequence Diffusion



Next-Token Prediction

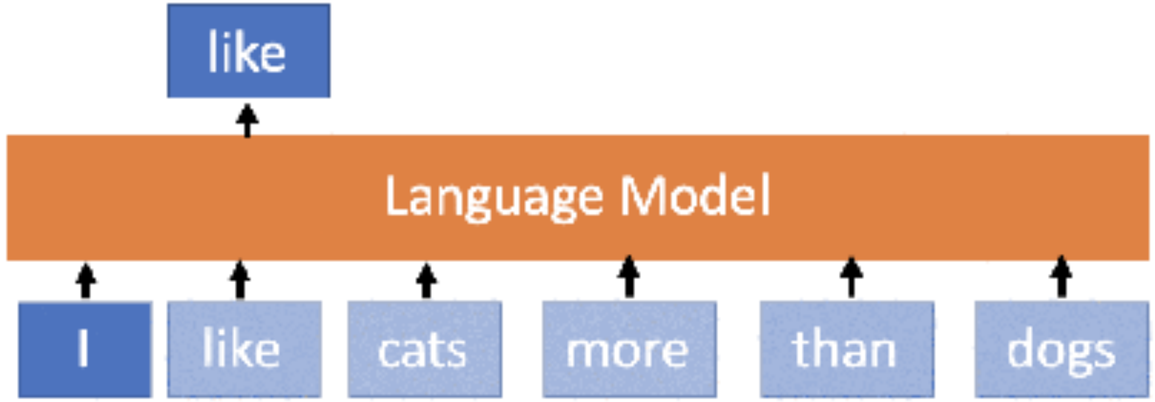


Two paradigms of sequence modeling

Full-Sequence Diffusion



Next-Token Prediction

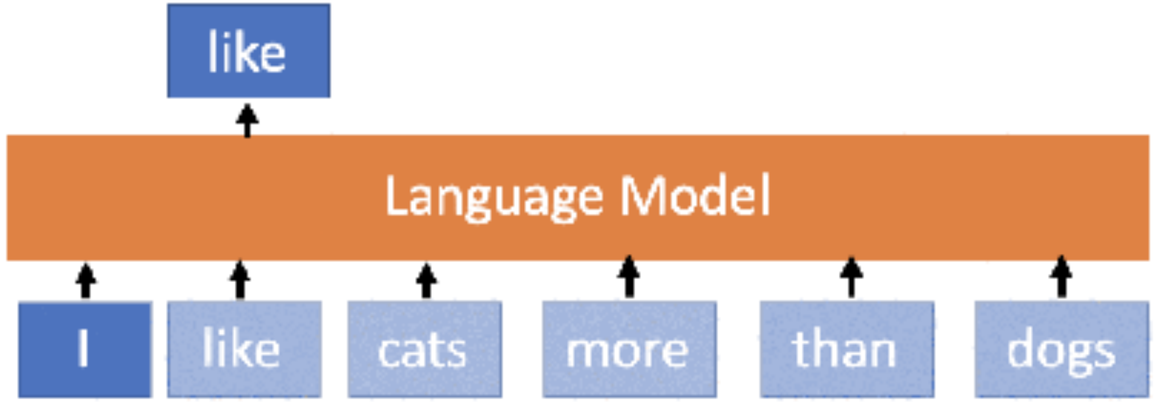


Two paradigms of sequence modeling

Video Models



LLMs

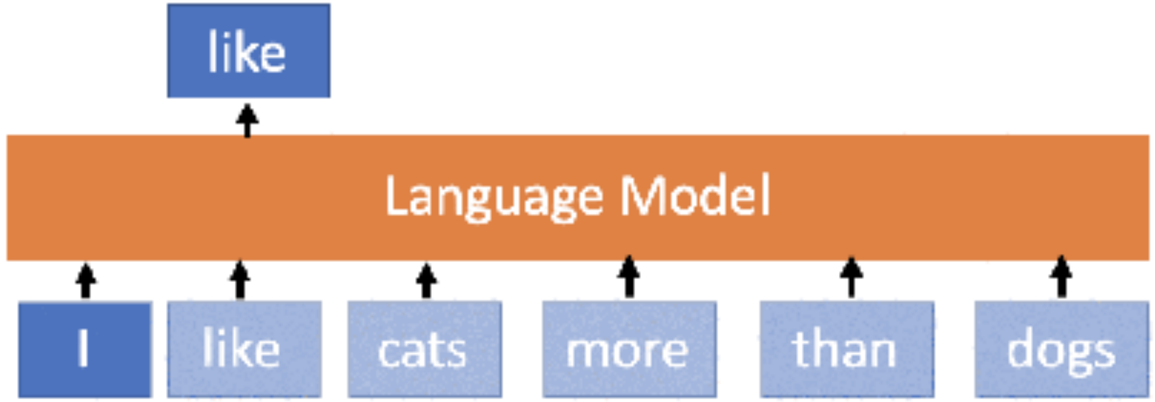


Two paradigms of sequence modeling

Video Models



LLMs

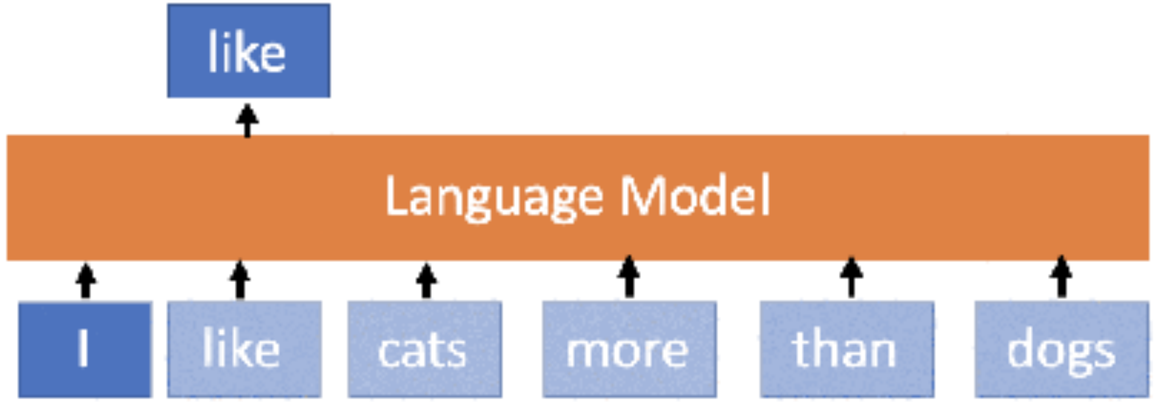


Two paradigms of sequence modeling

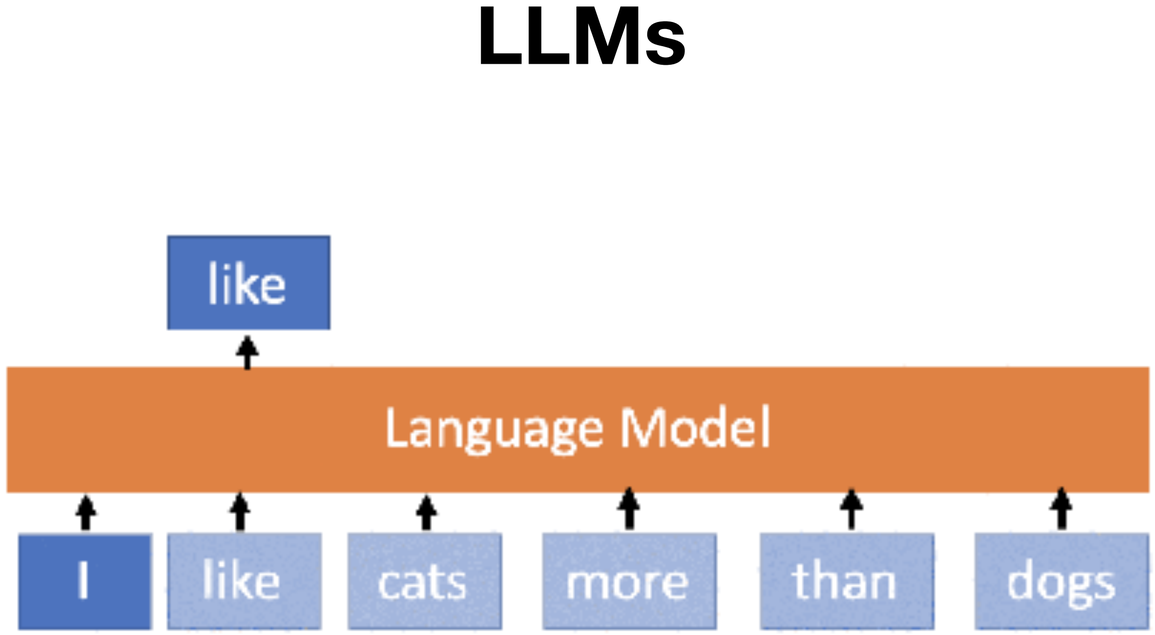
Video Models



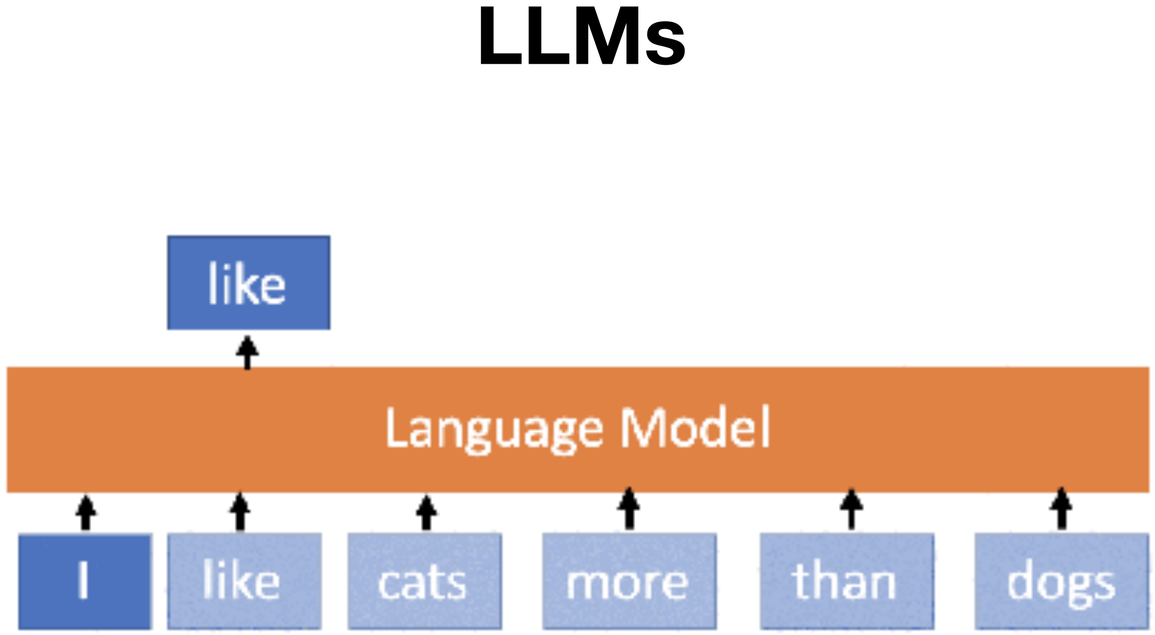
LLMs



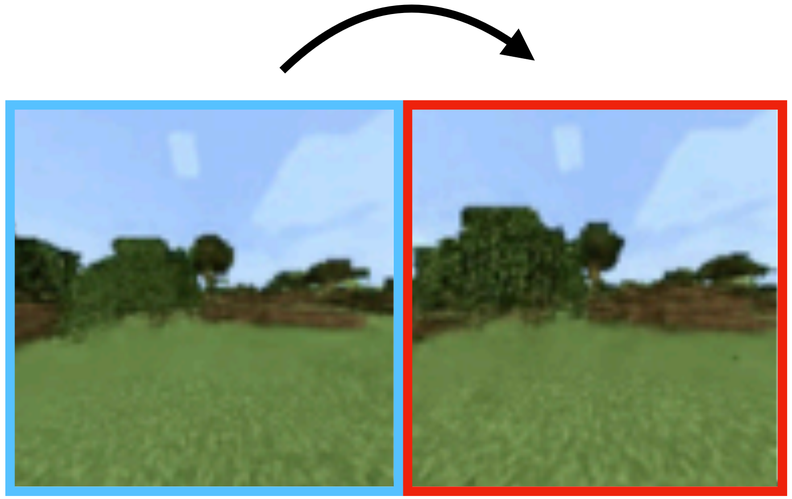
LLMs: next-token prediction



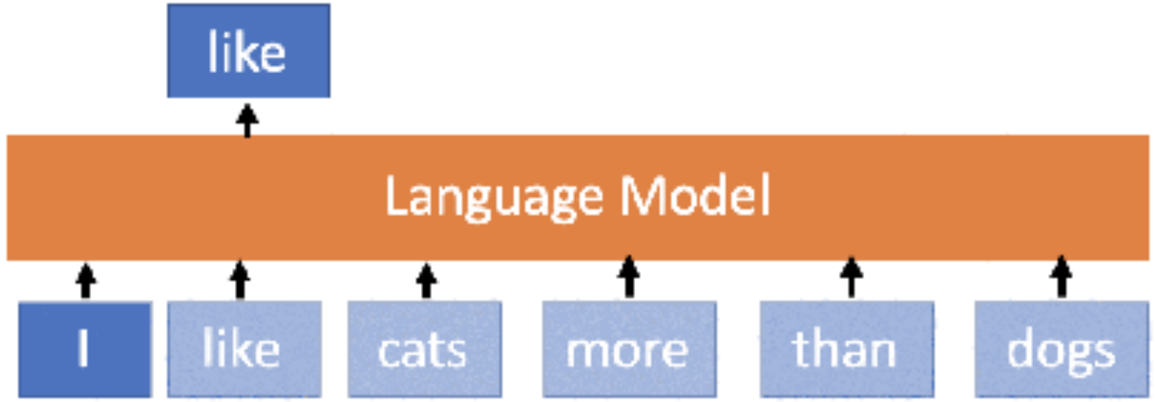
LLMs: next-token prediction



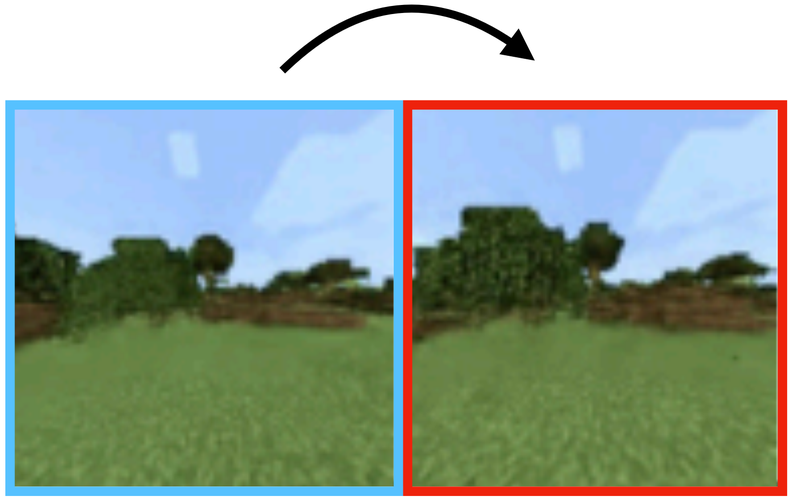
LLMs: next-token prediction



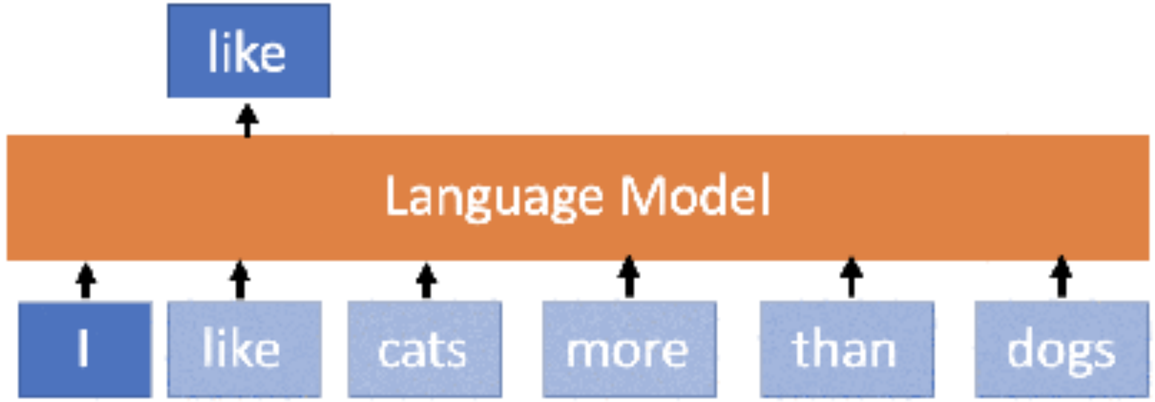
LLMs



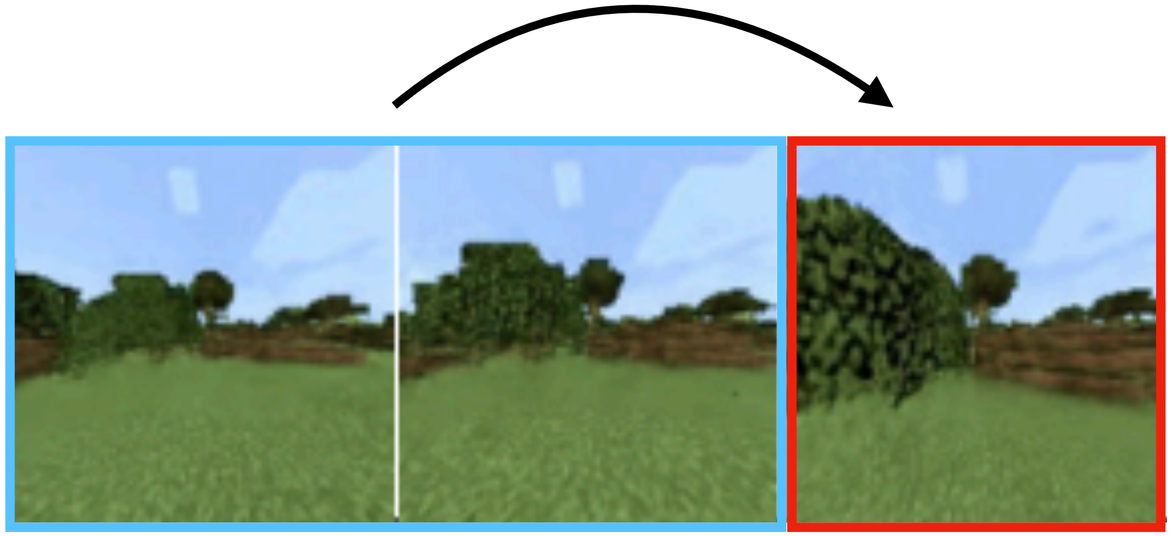
LLMs: next-token prediction



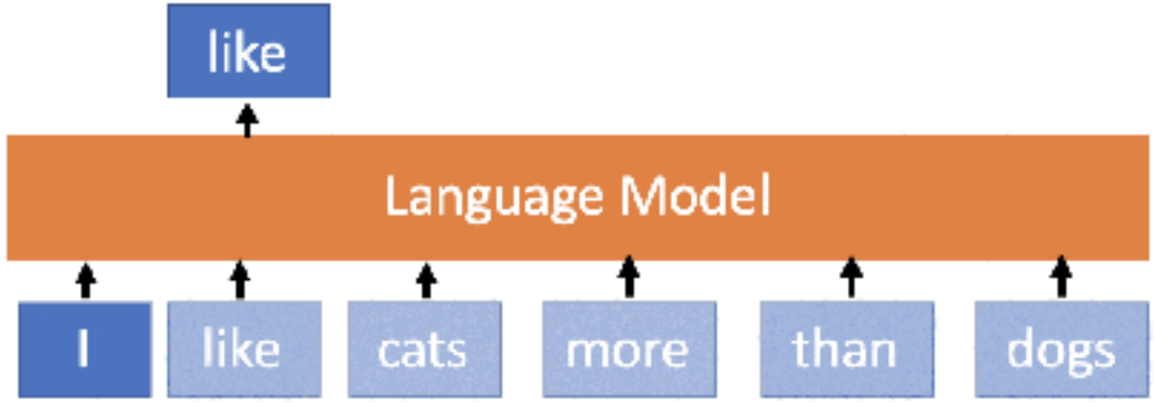
LLMs



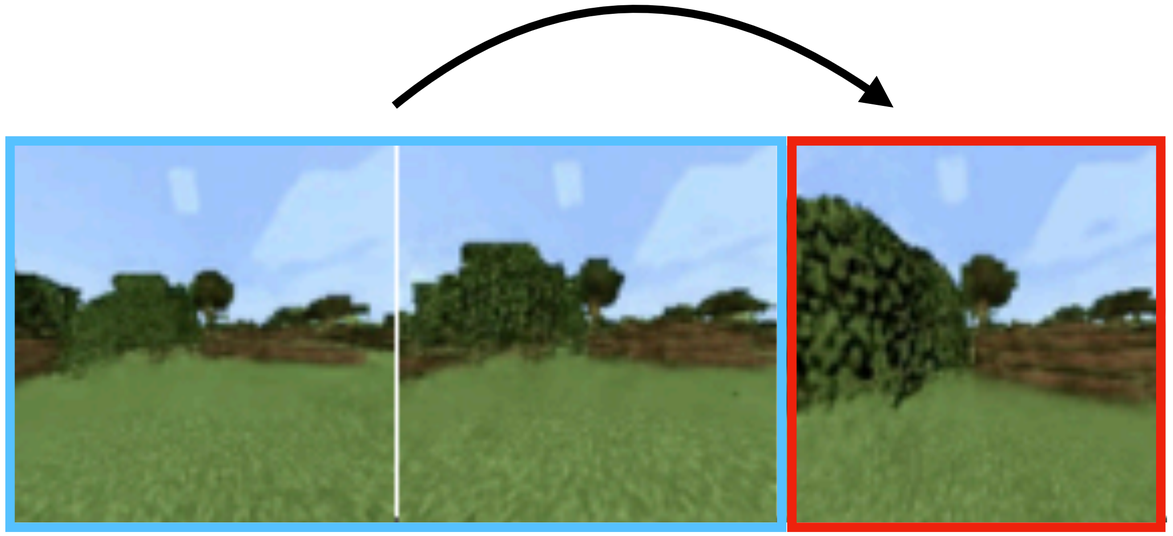
LLMs: next-token prediction



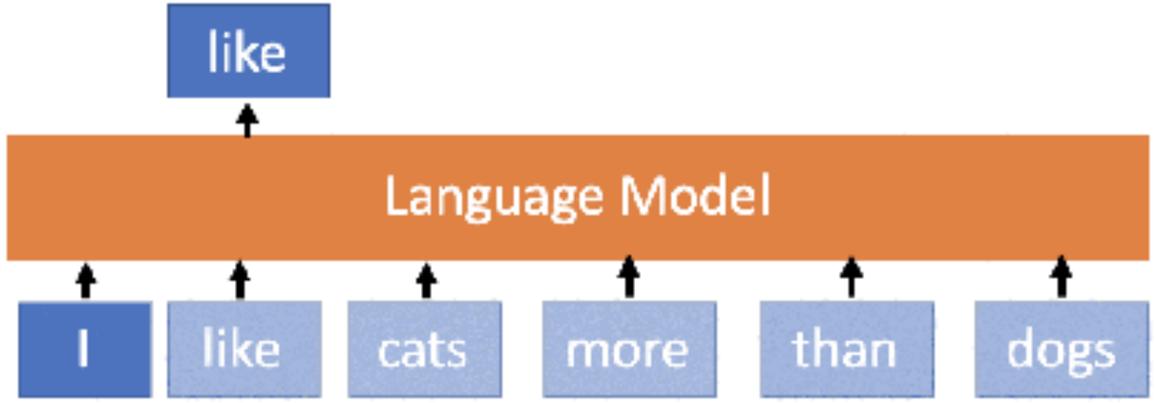
LLMs



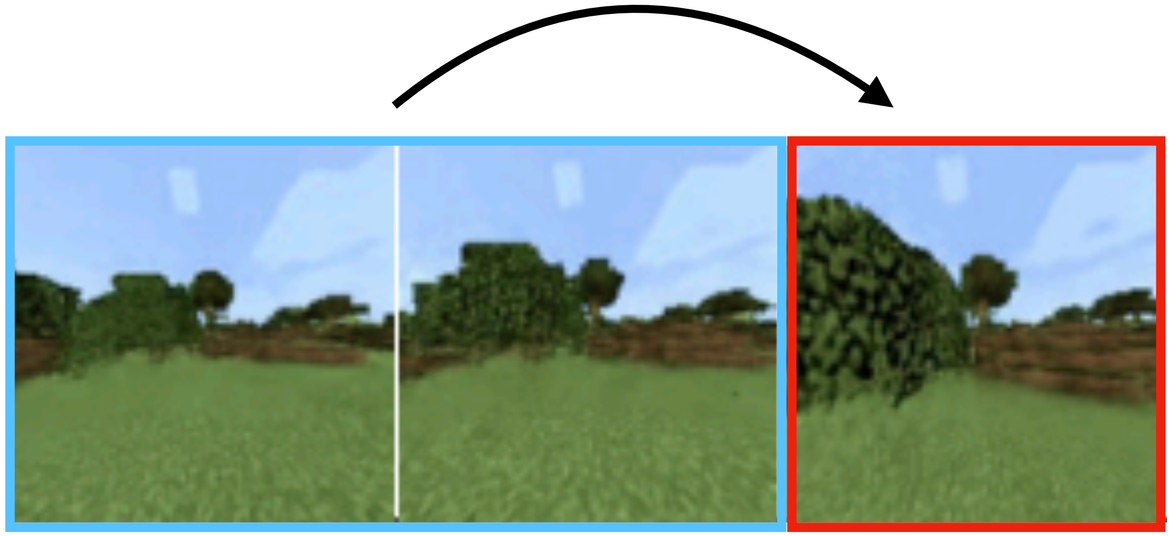
LLMs: next-token prediction



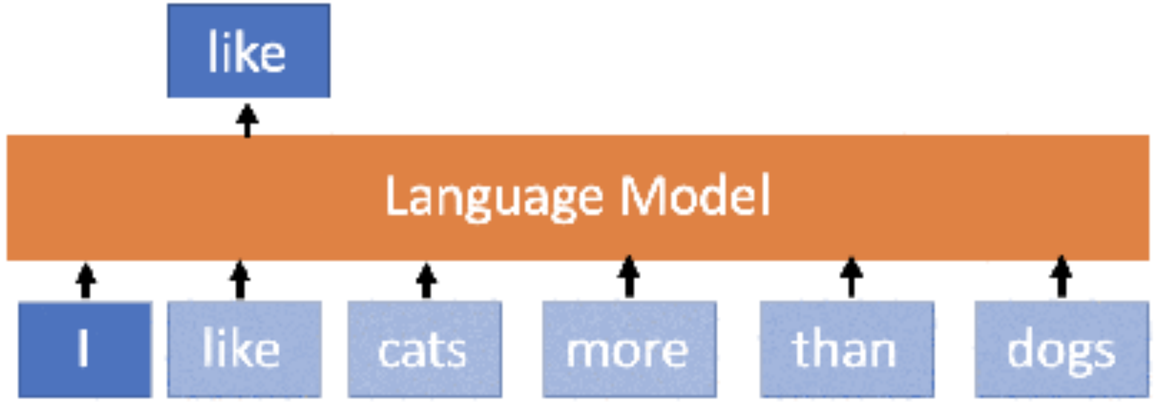
LLMs



LLMs: Do not work for continuous data



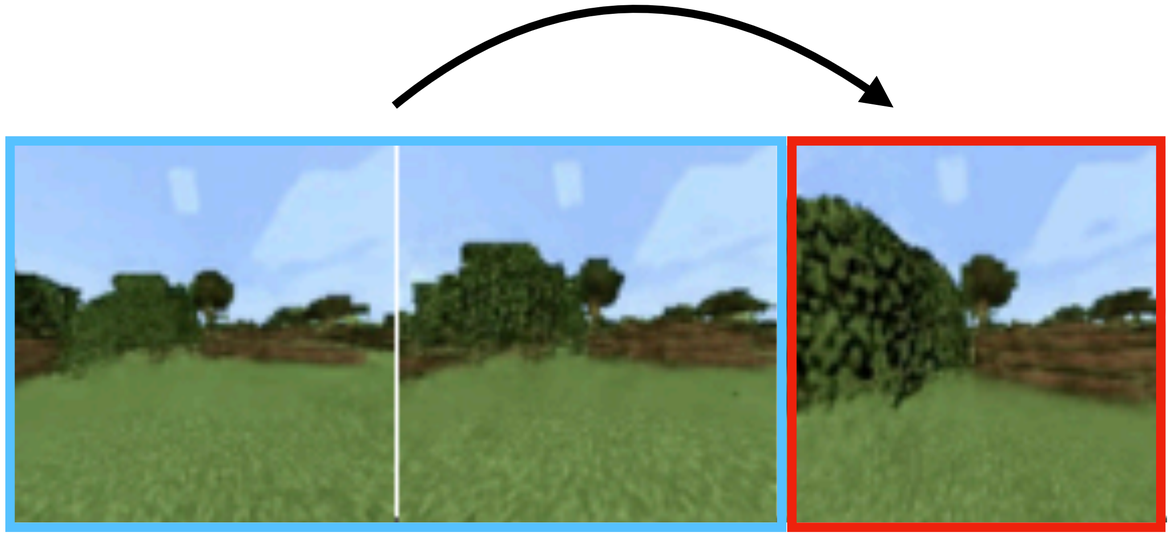
LLMs



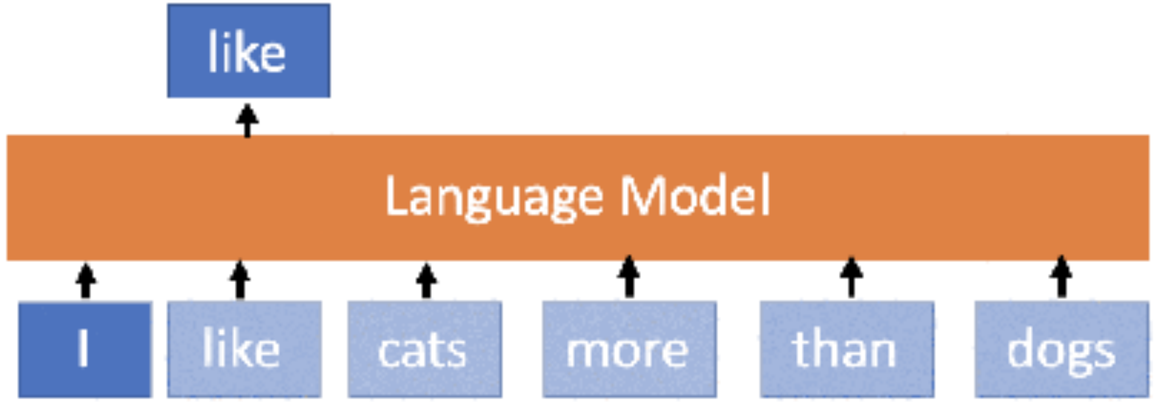
Auto-regression for long seq.

Text only, “Wrong Language”

LLMs: Do not work for continuous data



LLMs



Auto-regression for long seq.

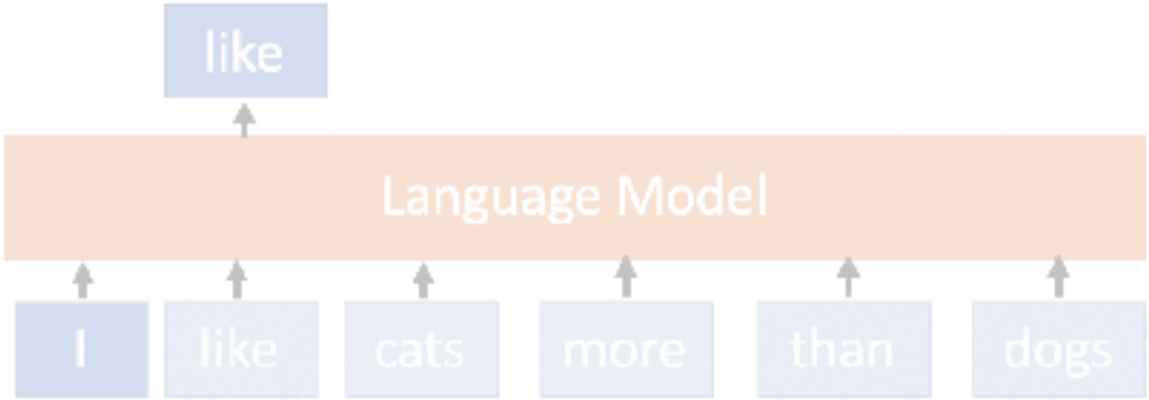
Text only, “Wrong Language”

To date: Two paradigms

Video Models



LLMs

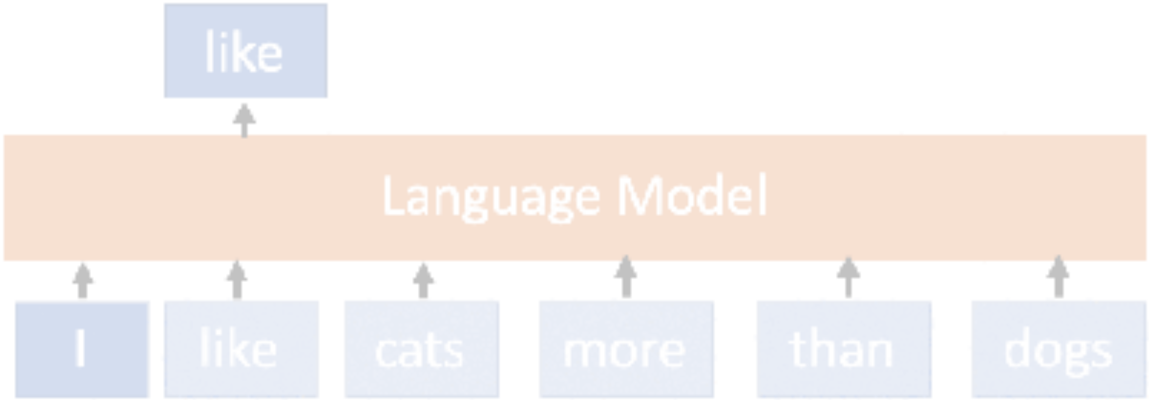


To date: Two paradigms

Video Models



LLMs

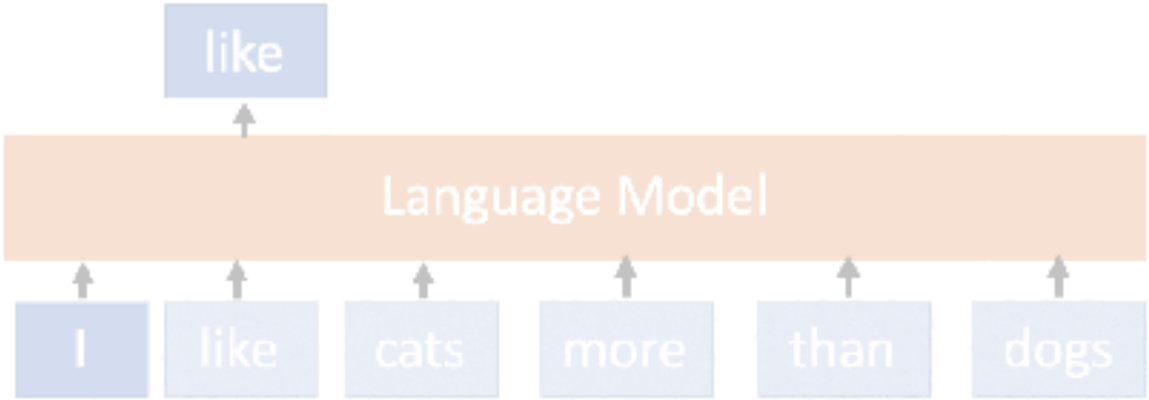


To date: Two paradigms

Video Models



LLMs



To date: Two paradigms

Video Models



To date: Two paradigms

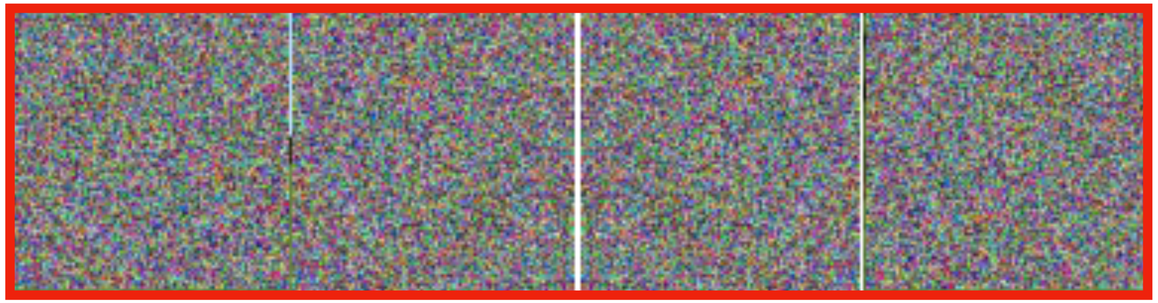
Video Models



Video Models: Full-Sequence Diffusion

Video Models

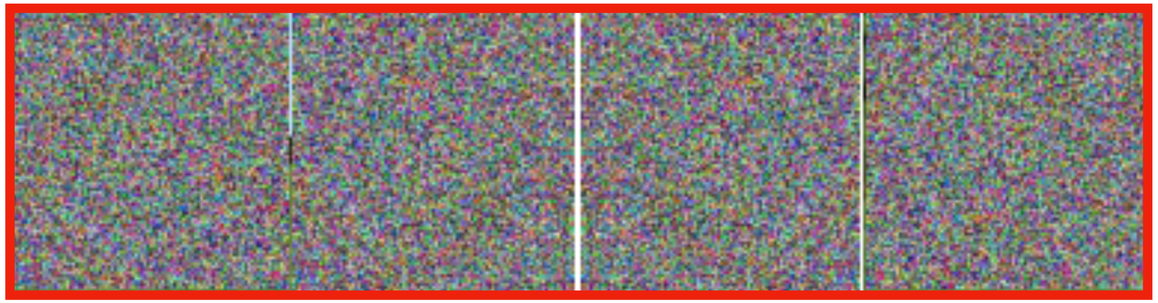




Video Models: Full-Sequence Diffusion

Video Models

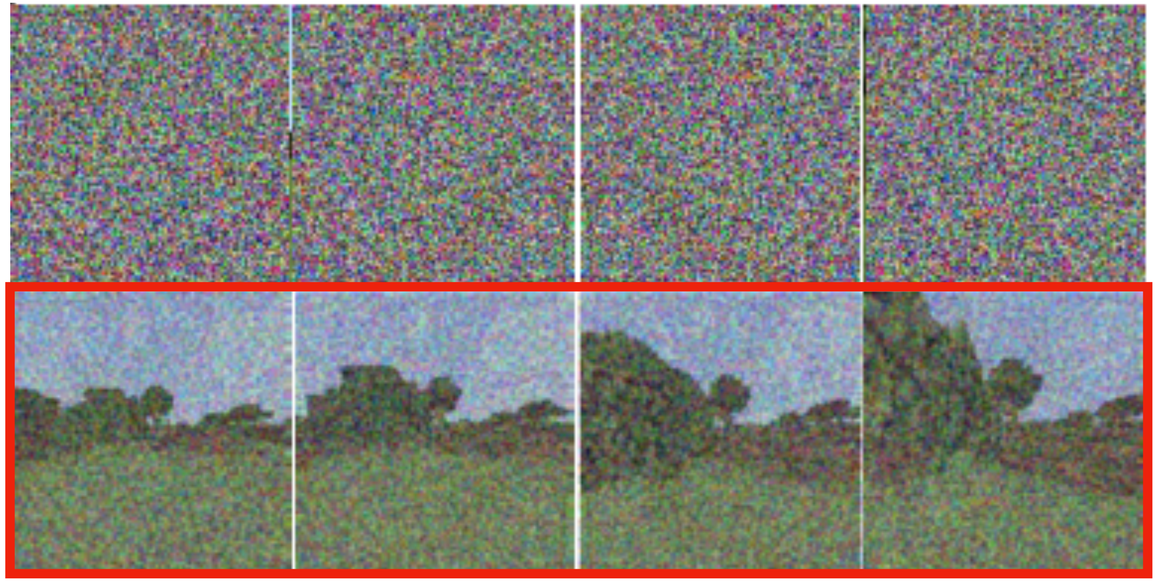




Video Models: Full-Sequence Diffusion

Video Models

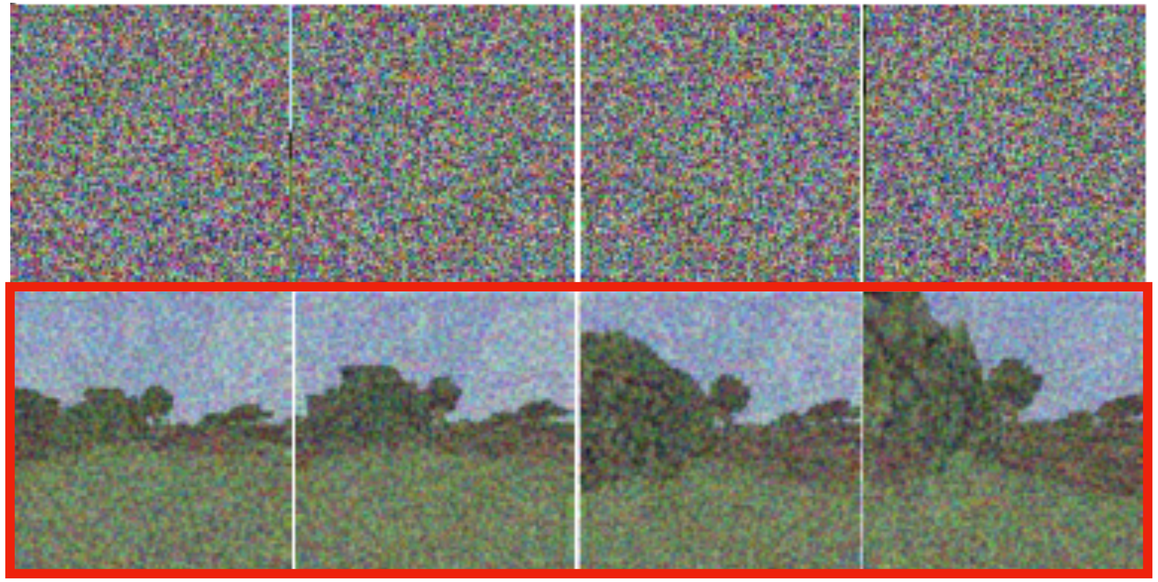




Video Models: Full-Sequence Diffusion

Video Models

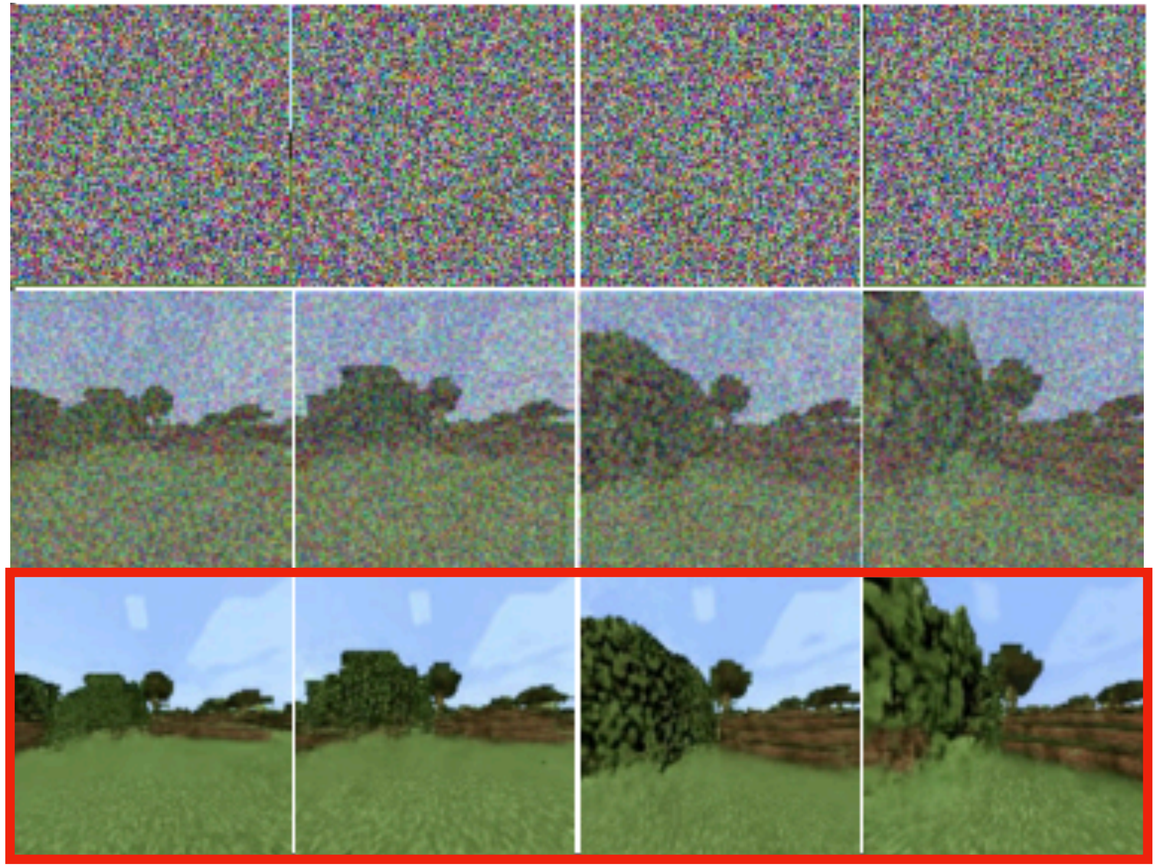




Video Models: Full-Sequence Diffusion

Video Models

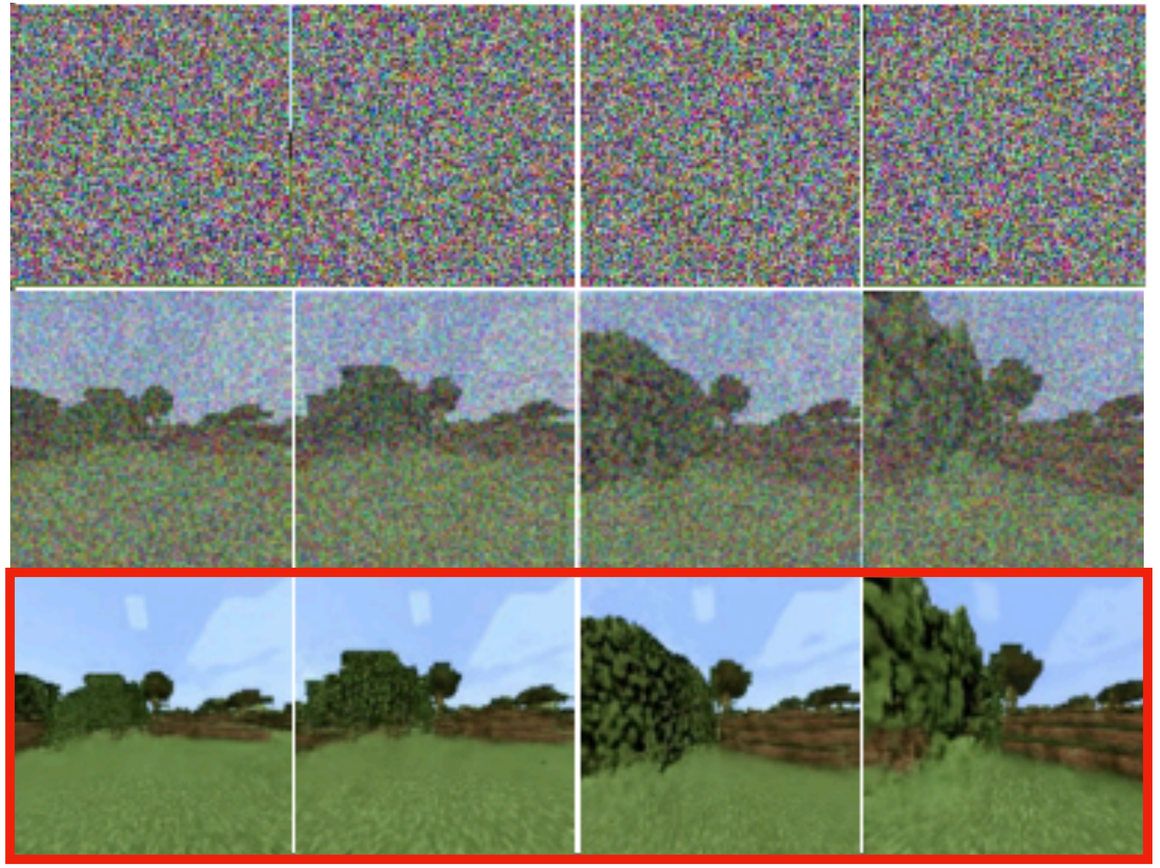




Video Models: Full-Sequence Diffusion

Video Models





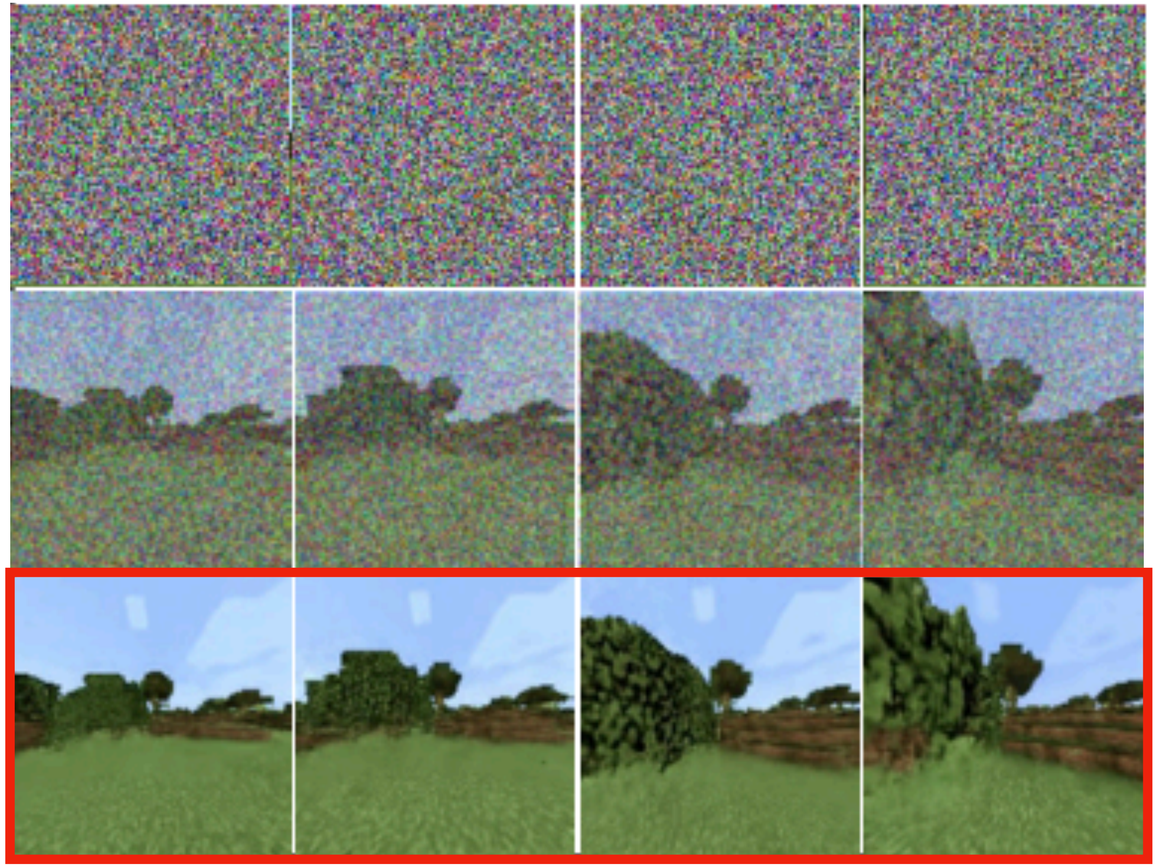
Video Models: Full-Sequence Diffusion

Video Models



Not interactive

Only Short Videos



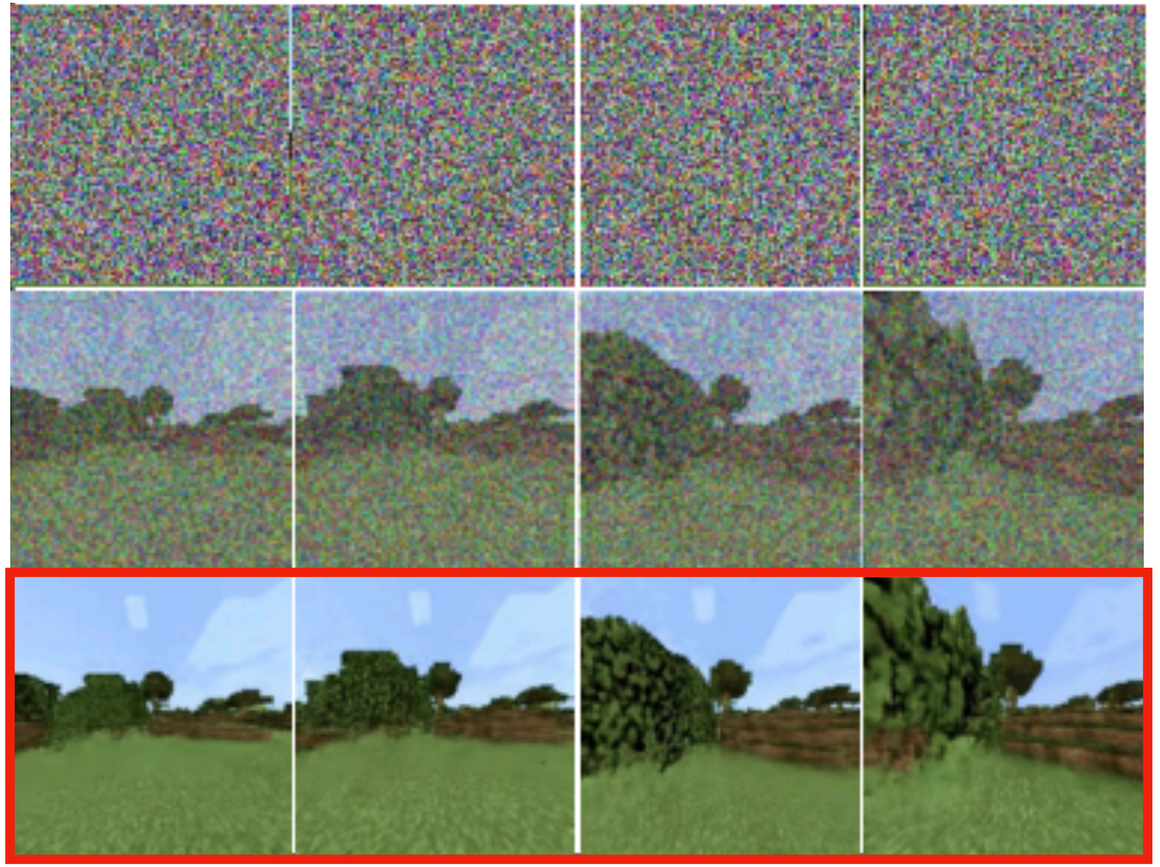
Video Models: Full-Sequence Diffusion

Video Models



Not interactive

Only Short Videos



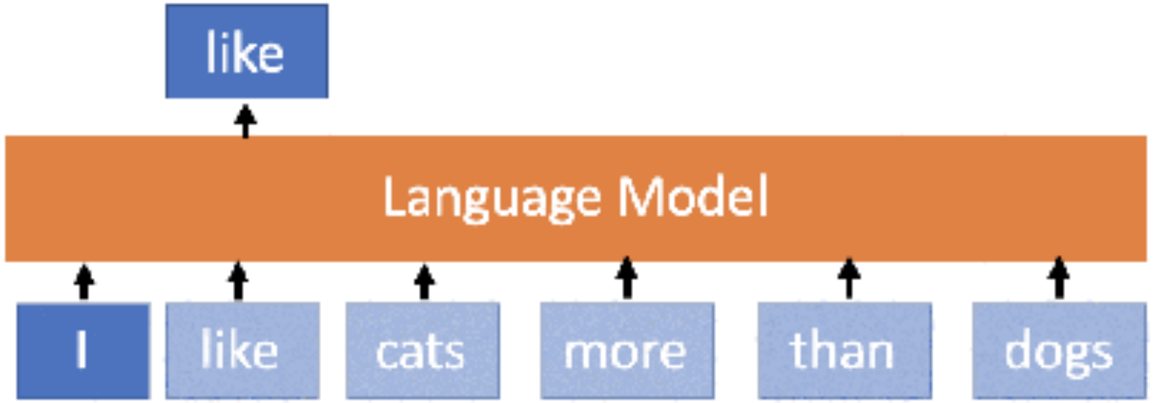
Video Models



Not interactive

Only Short Videos

LLMS



Long Sequences

Text only, “Wrong Language”

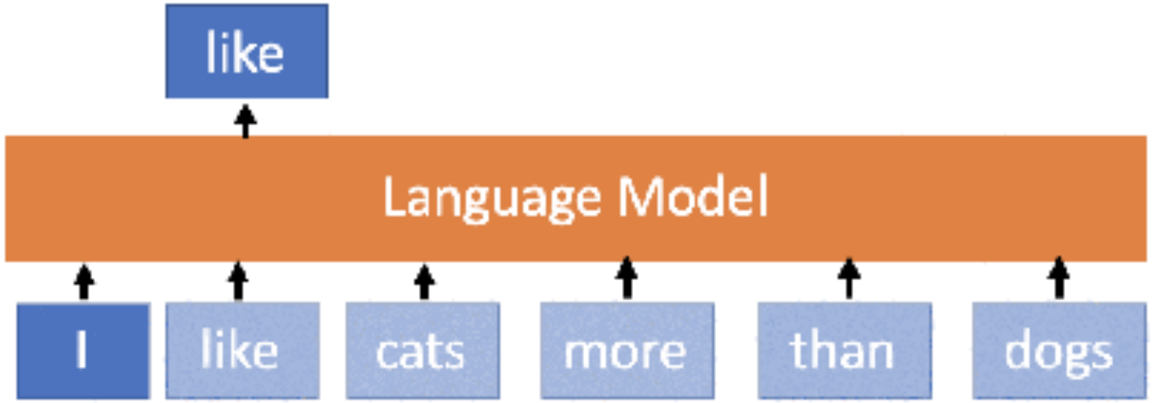
Video Models



Not interactive

Only Short Videos

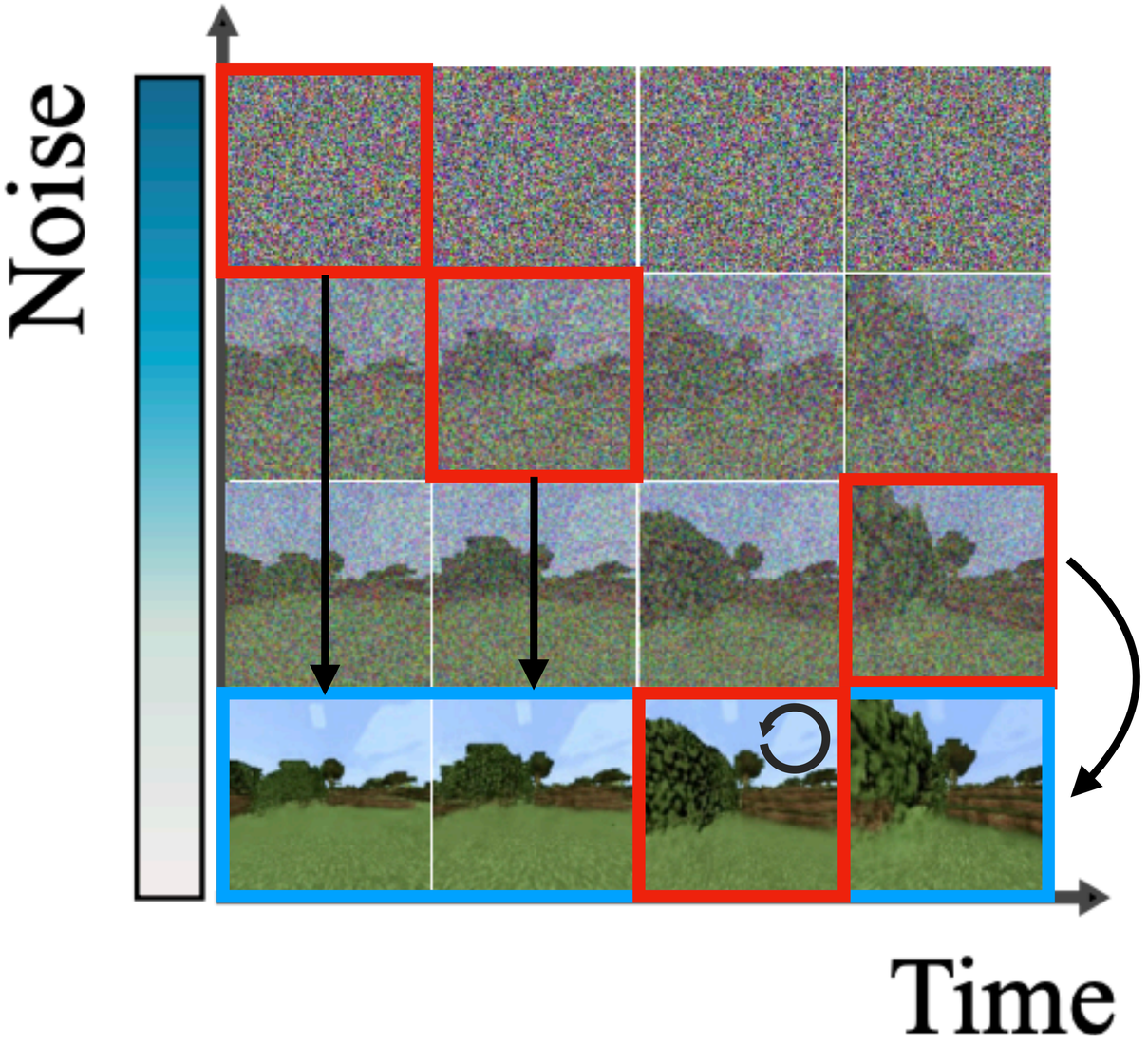
LLMS



Long Sequences

Text only, “Wrong Language”

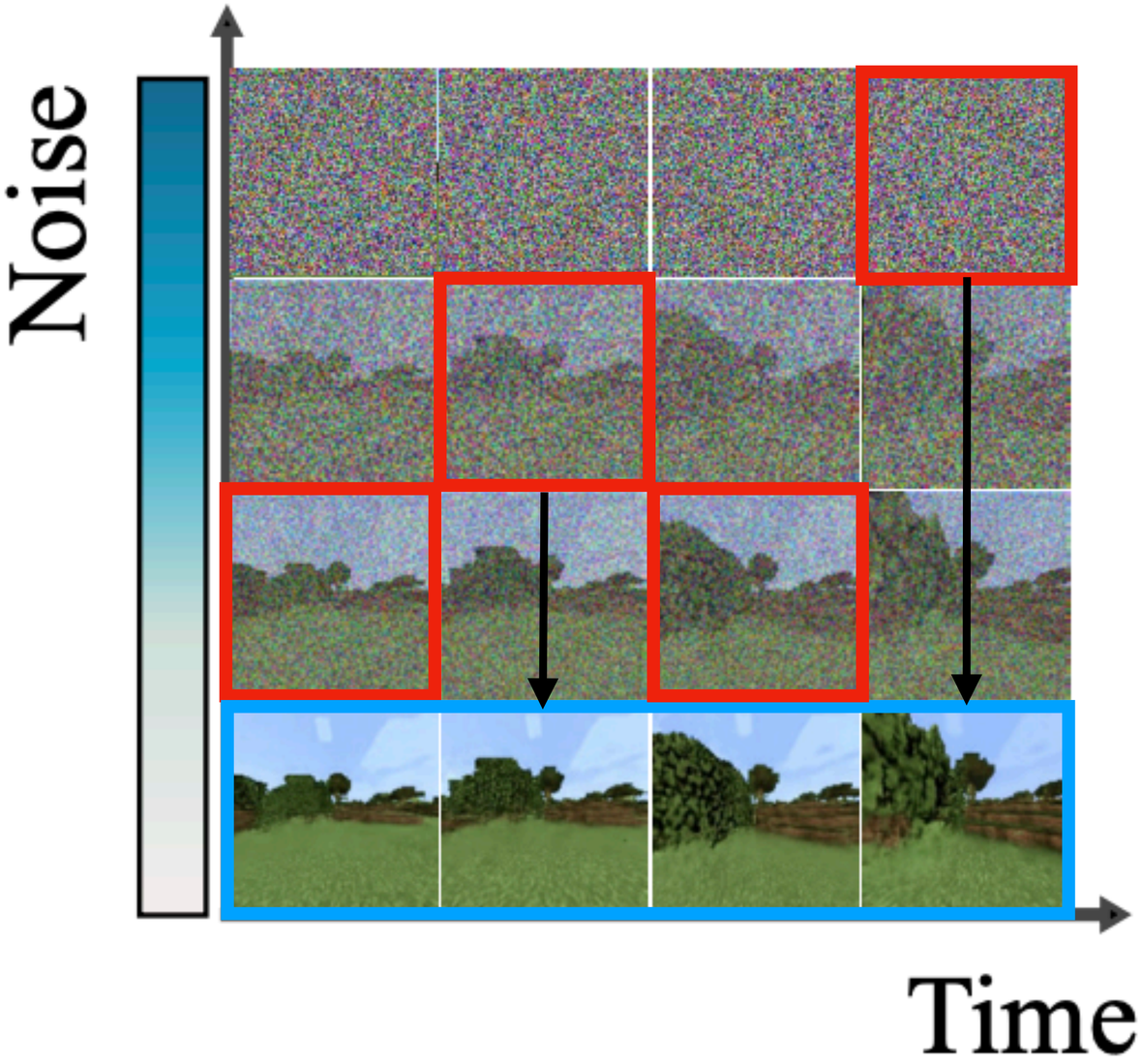
Diffusion Forcing



Key Idea: Train the model to denoise frames with *random combinations of noise levels!*

[Chen et al. 2024, Song and Chen et al. 2024]

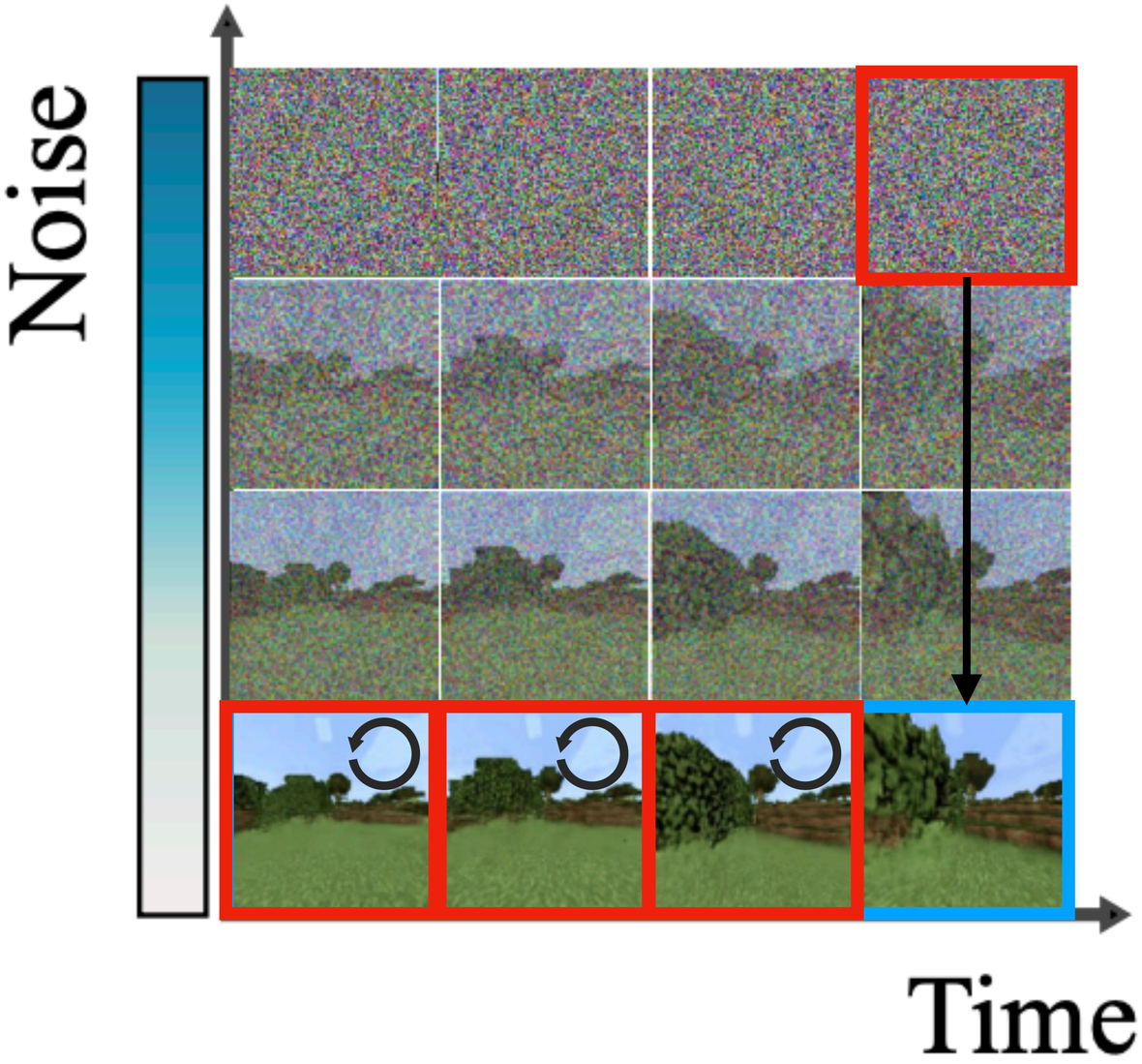
Diffusion Forcing



Key Idea: Train the model to denoise frames with *random combinations of noise levels!*

[Chen et al. 2024, Song and Chen et al. 2024]

Diffusion Forcing

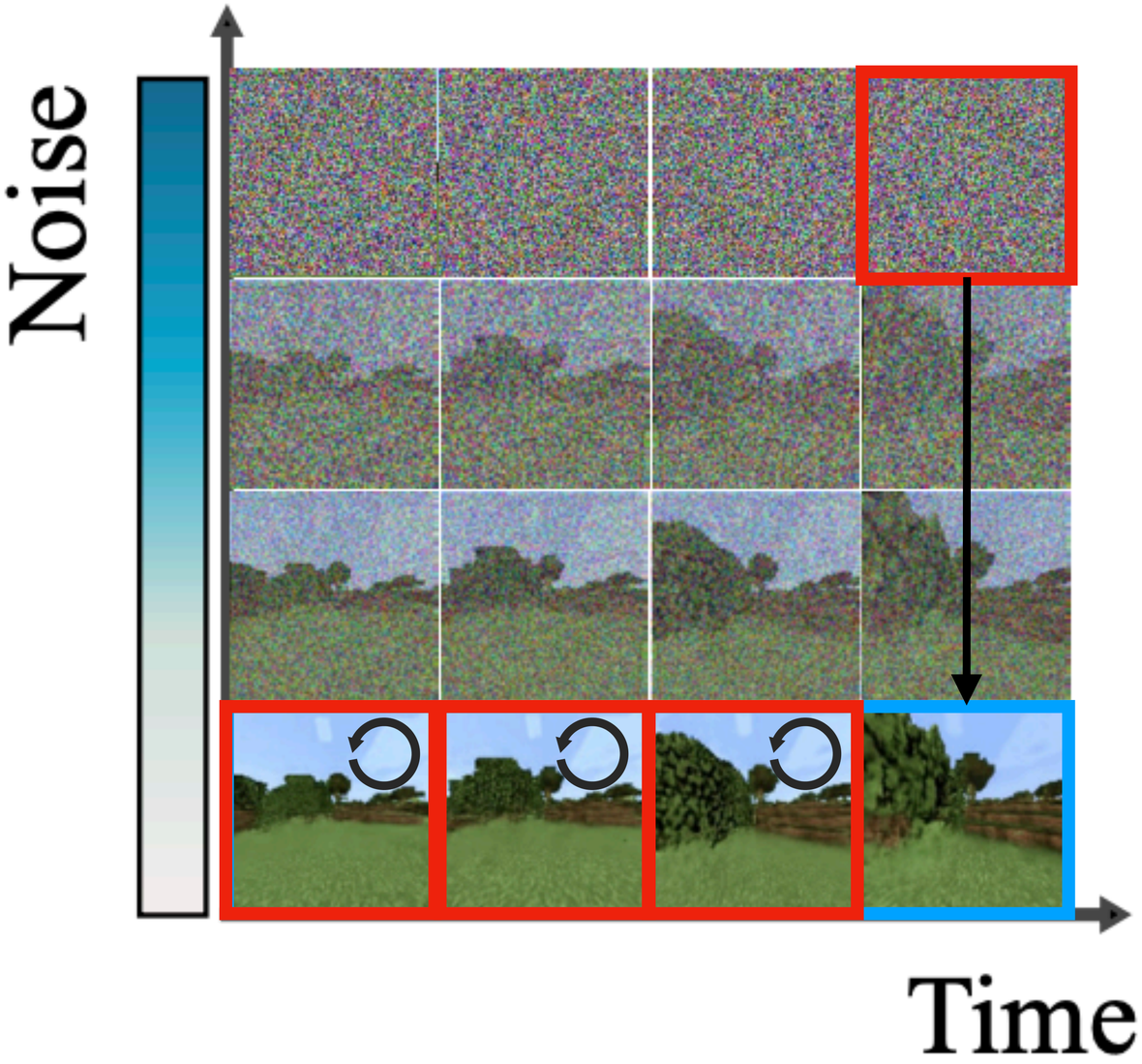


Key Idea: Train the model to denoise frames with *random combinations of noise levels!*

Over time, the model sees *every possible combination of noise levels*

[Chen et al. 2024, Song and Chen et al. 2024]

Diffusion Forcing



Key Idea: Train the model to denoise frames with *random combinations of noise levels!*

Over time, the model sees *every possible combination of noise levels*

This generalizes both auto-regressive sampling and full-sequence diffusion

[Chen et al. 2024, Song and Chen et al. 2024]

Diffusion Forcing (Chen et al., 2024)

"LLM-style"

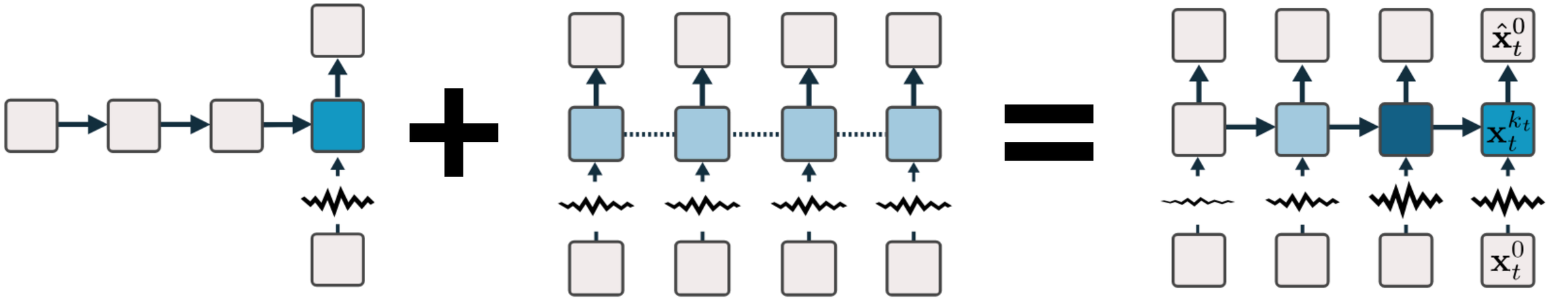
Teacher Forcing

"Video-gen style"

Full-Seq. Diffusion

"World Model"

Diffusion Forcing



Diffusion Forcing: **Interactive, Long-Form** World Model

Rollouts with an RNN

"LLM-style"



compounding error
(exposure bias)

"Video-gen style"



consistency issues

Diffusion Forcing



Diffusion Forcing: **Interactive, Long-Form** World Model

Rollouts with an RNN

"LLM-style"



compounding error
(exposure bias)

"Video-gen style"



consistency issues

Diffusion Forcing



Diffusion Forcing: **Interactive, Long-Form** World Model

Rollouts with an RNN

"LLM-style"



compounding error
(exposure bias)

"Video-gen style"



consistency issues

Diffusion Forcing



Diffusion Forcing: **Interactive, Long-Form** World Model

Rollouts with an RNN

"LLM-style"



compounding error
(exposure bias)

"Video-gen style"



consistency issues

Diffusion Forcing



October 31, 2024

Oasis: A Universe in a Transformer

Decart, Etched



▷ Try demo

↗ Decart Blog | View code | Model weights | ↗ Etched Blog

We're excited to announce Oasis, the first experiential, realtime, open-world AI model. It's an interactive video experience, but entirely generated by AI. Oasis is the first step in our research towards more complex interactive worlds.

Oasis takes in user keyboard input and generates a real-time experience, including physics, rules, and graphics. You can move around, jump, pick up items, break blocks, and more. There is no physics engine; just a foundation model.

We believe fast transformer inference is the missing link to making generative video a reality. Using Decart's inference engine, we show that real-time video is possible. When Etched's transformer ASIC, Sohu, is released, we can run models like Oasis in 4K. Today, we're releasing Oasis's code, the weights of a 500M parameter model you can run locally, and a live demo of a larger checkpoint.

October 31, 2024

Oasis: A Universe in a Transformer

Decart, Etched



▷ Try demo

↗ Decart Blog | View code | Model weights | ↗ Etched Blog

We're excited to announce Oasis, the first experiential, realtime, open-world AI model. It's an interactive video experience, but entirely generated by AI. Oasis is the first step in our research towards more complex interactive worlds.

Oasis takes in user keyboard input and generates a real-time experience, including physics, rules, and graphics. You can move around, jump, pick up items, break blocks, and more. There is no physics engine; just a foundation model.

We believe fast transformer inference is the missing link to making generative video a reality. Using Decart's inference engine, we show that real-time video is possible. When Etched's transformer ASIC, Sohu, is released, we can run models like Oasis in 4K. Today, we're releasing Oasis's code, the weights of a 500M parameter model you can run locally, and a live demo of a larger checkpoint.

October 31, 2024

Oasis: A Universe in a Transformer

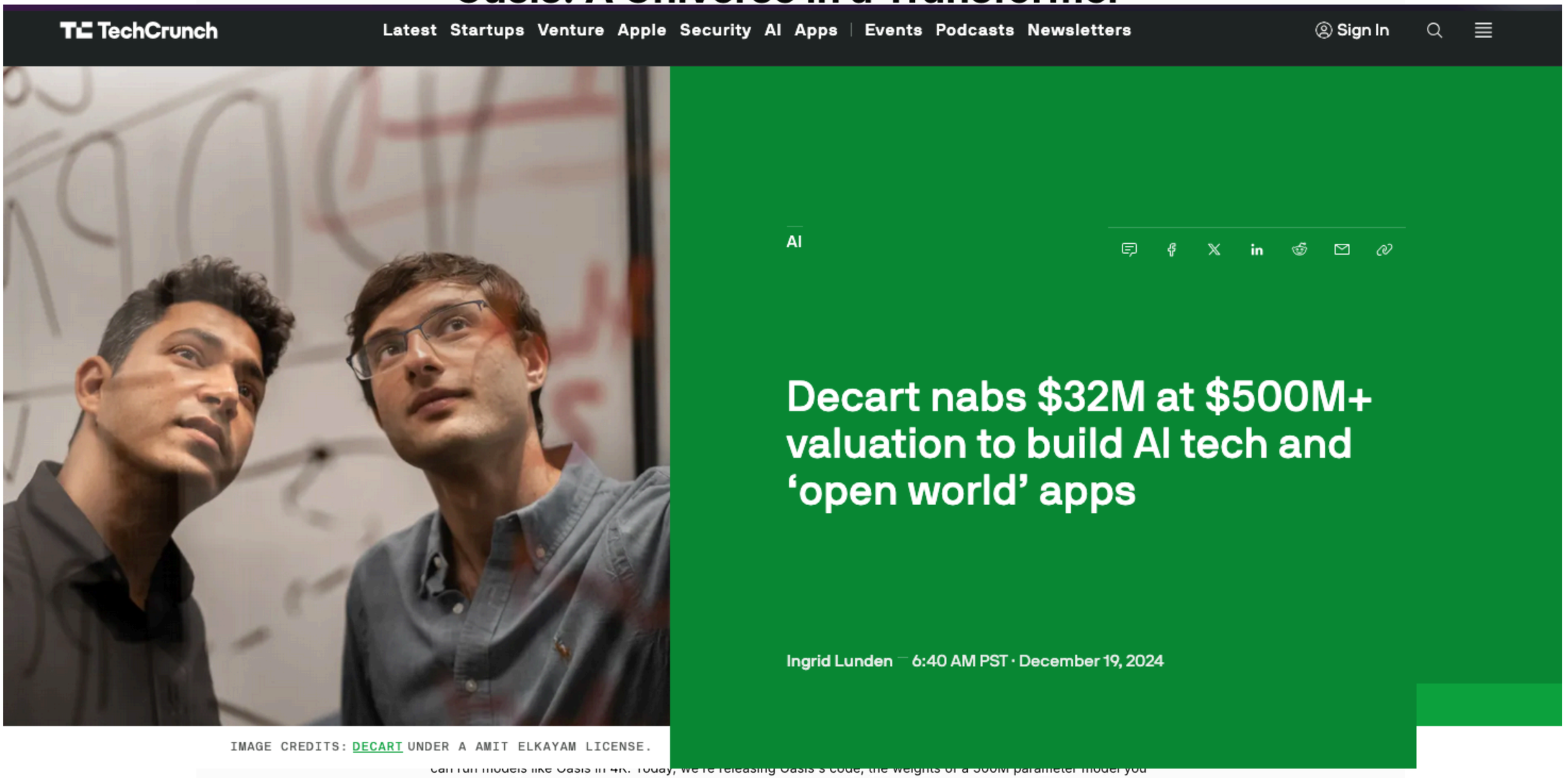


IMAGE CREDITS: DECART UNDER A AMIT ELKAYAM LICENSE.

can run models like Oasis in 4K. Today, we're releasing Oasis's code, the weights of a 500M parameter model you can run locally, and a live demo of a larger checkpoint.



Scaling up: February 2025



Scaling up: February 2025

August 2025: Genie 3



August 2025: Genie 3



September 2025: DeepMind's Dreamer 4

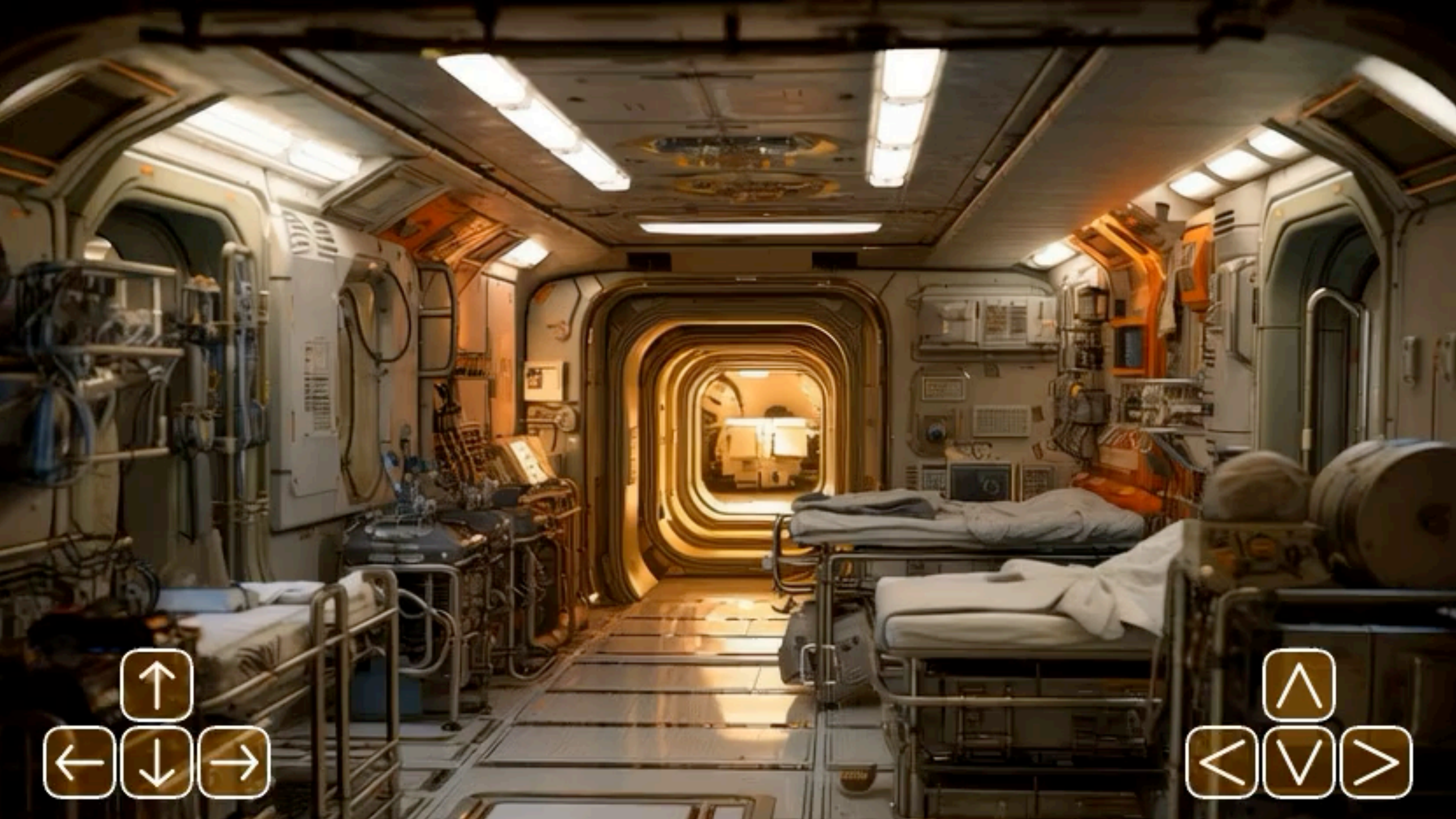


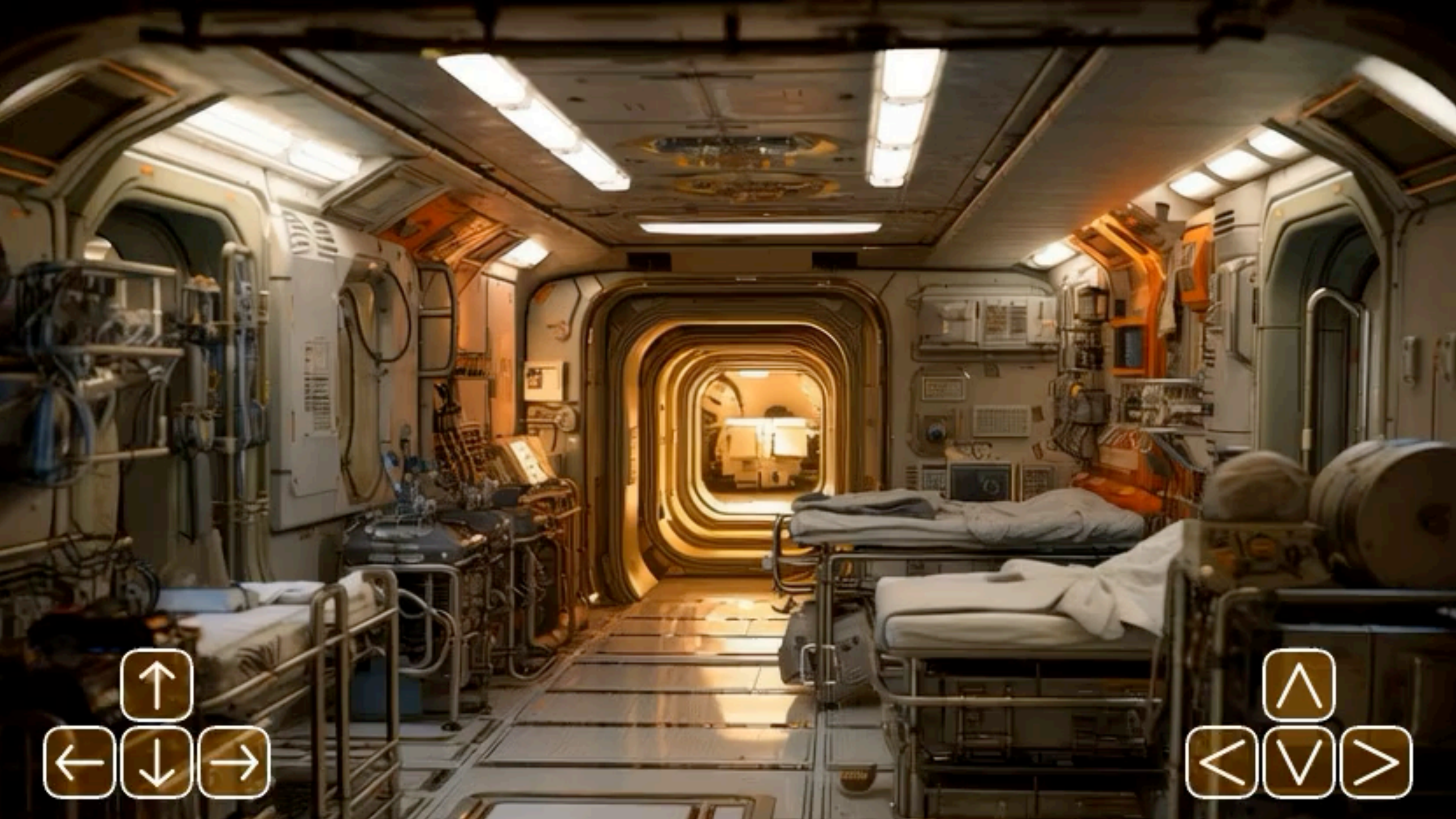
Trains an agent in an imagined world built with Diffusion Forcing

September 2025: DeepMind's Dreamer 4



Trains an agent in an imagined world built with Diffusion Forcing





December 2025: Adobe Relic



December 2025: Adobe Relic

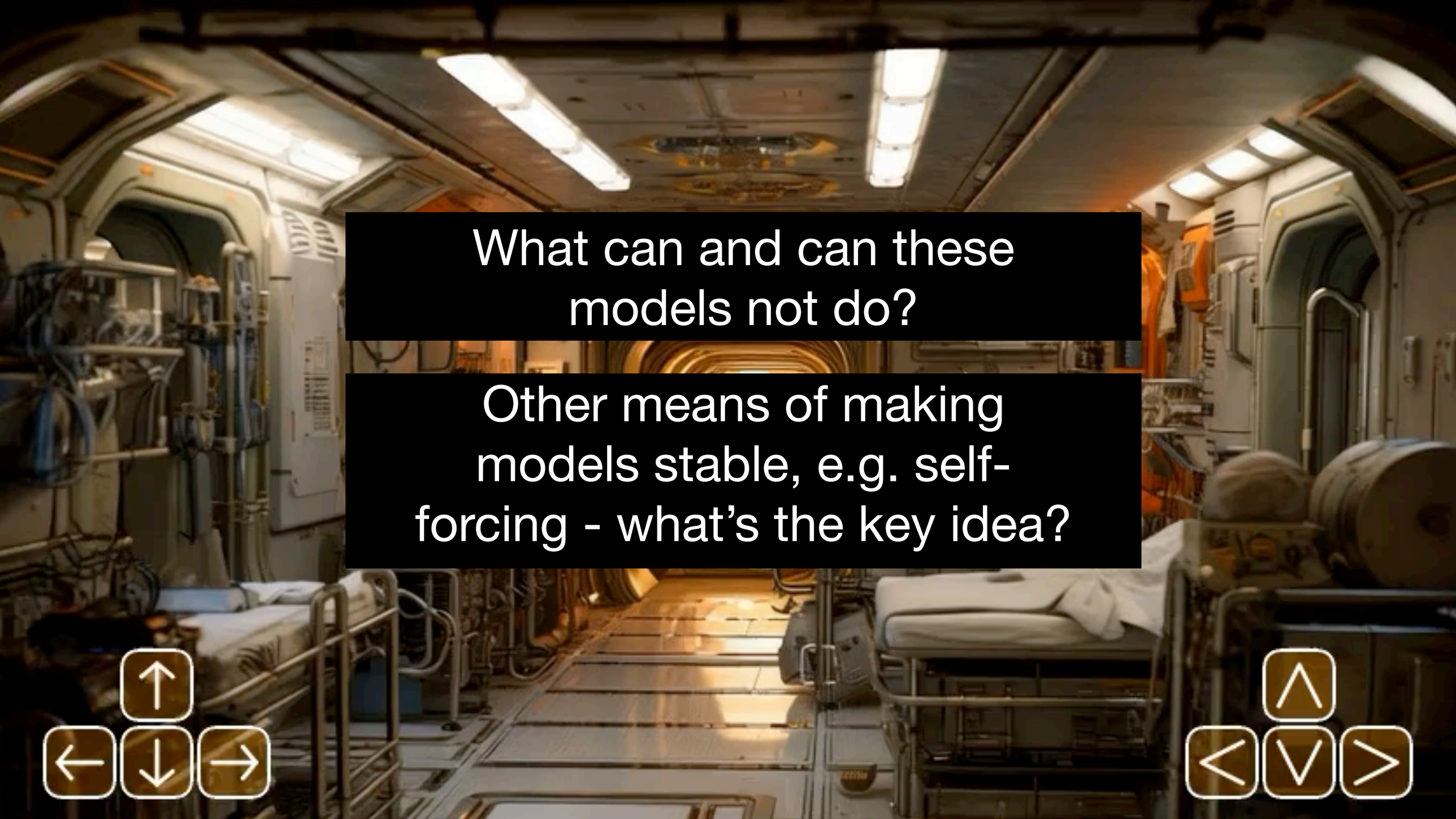


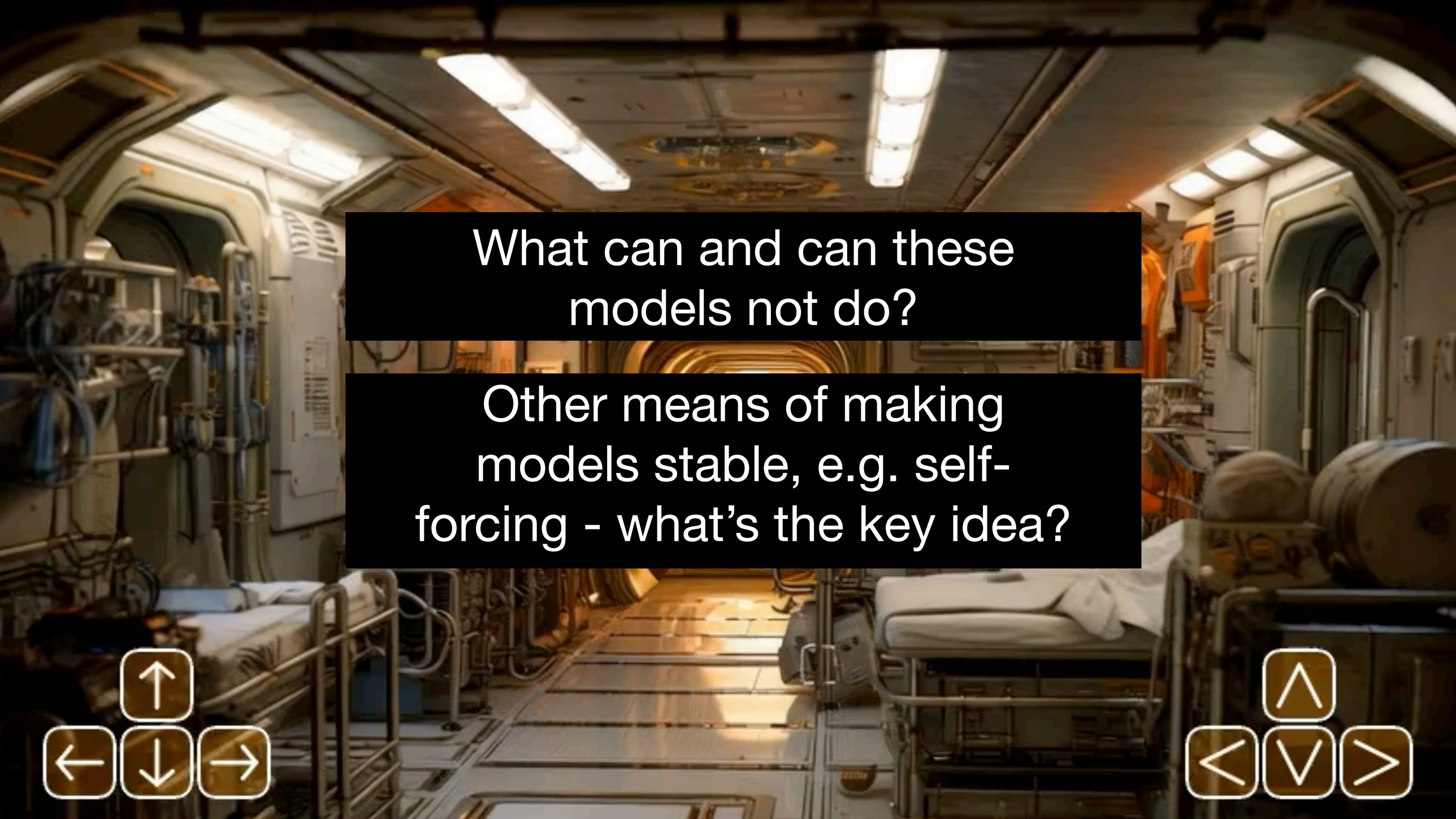
Trained w/ Diffusion Forcing!

December 2025: Adobe Relic



Trained w/ Diffusion Forcing!


What can and can these models not do?
Other means of making models stable, e.g. self-forcing - what’s the key idea?


What can and can these models not do?
Other means of making models stable, e.g. self-forcing - what’s the key idea?

Generative modeling: Just another pretext task for Representation Learning?

Pretext task:

Spatial imputation

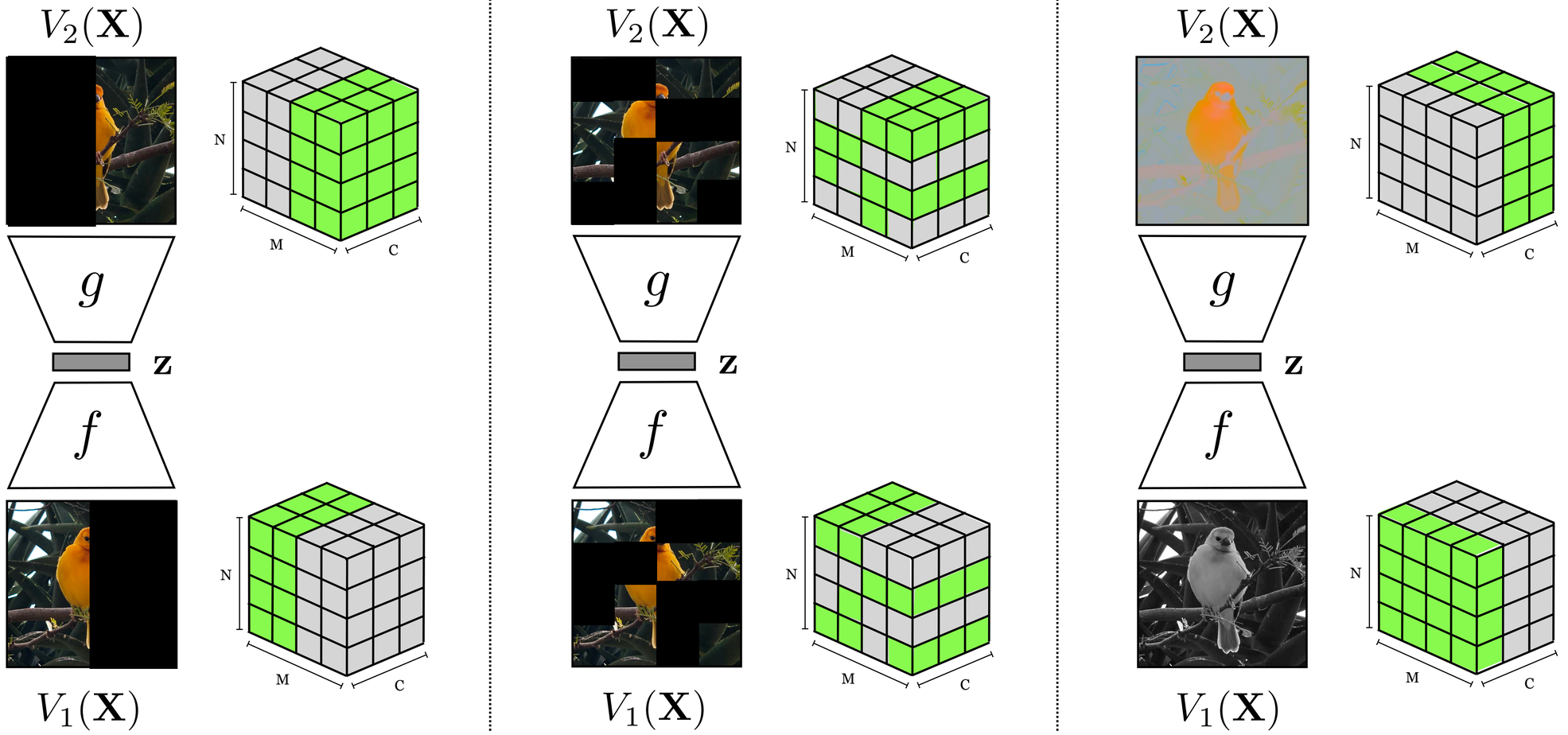
Spatial imputation

Channel imputation

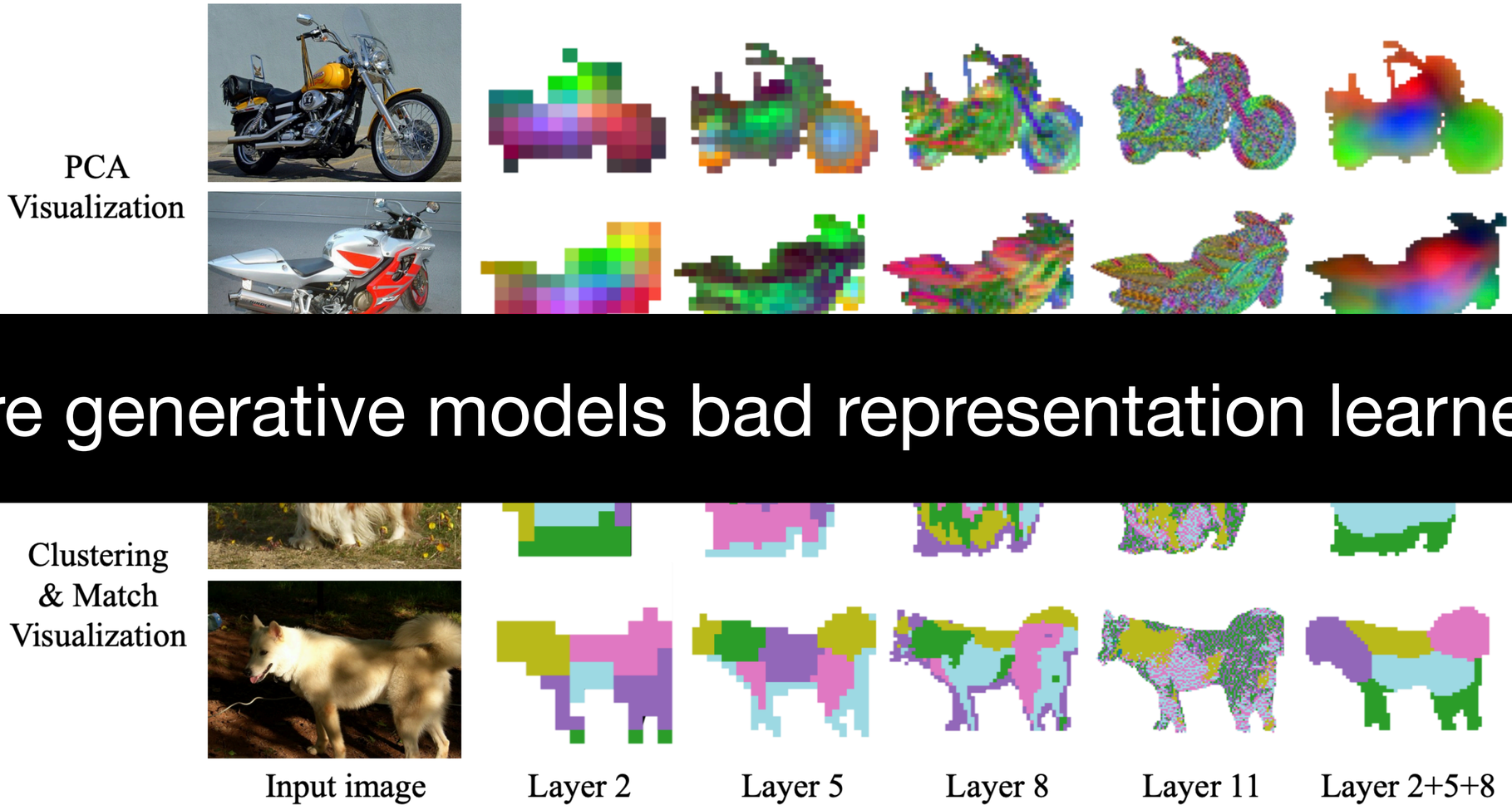
Model schematic:

Observed

Masked



While latent representations of generative models have *some* info, they fall far behind of dedicated SSL systems



Are generative models bad representation learners?

Figure 2: **Analysis of features from different decoder layers in SD.**

Zhang et al., A Tale of Two Features: Stable Diffusion Complements DINO for Zero-Shot Semantic Correspondence, 2023

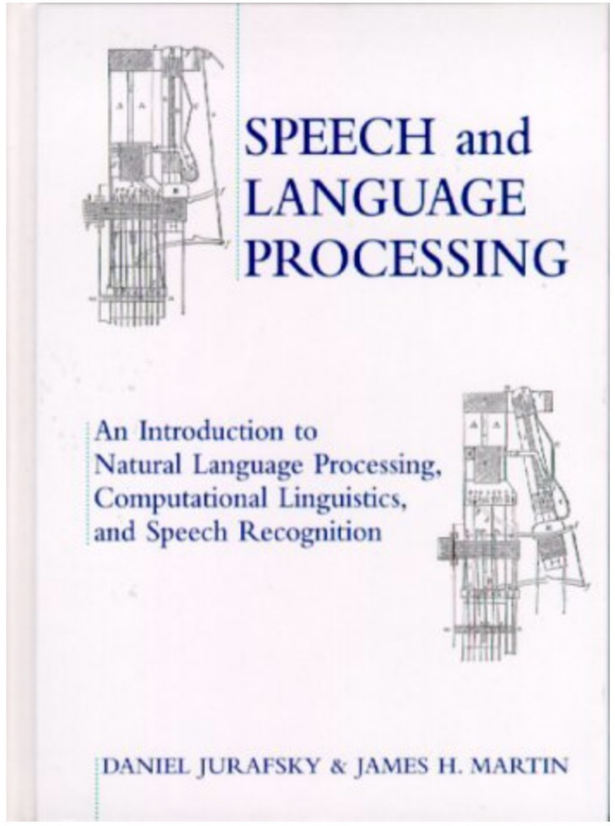
Let’s take some inspiration from NLP…

- In NLP, any and all benchmarks are now dominated by fine-tuned LLMs.
- LLMs are just a generative model: the split into representation learning / generative modeling does not exist!
- This is *not* the case in vision: there is seemingly no one-size-fits-all generative model we can train that solves all the vision tasks (3D reconstruction, classification, video generation, image editing…)
- Let’s reflect on that.

One Possibility: we are testing for the *wrong thing*



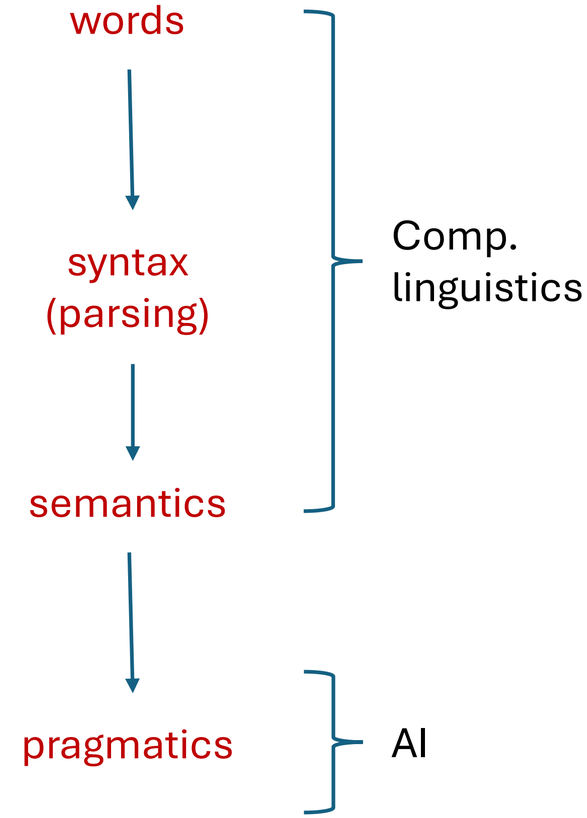
NLP circa 1999



1st edition

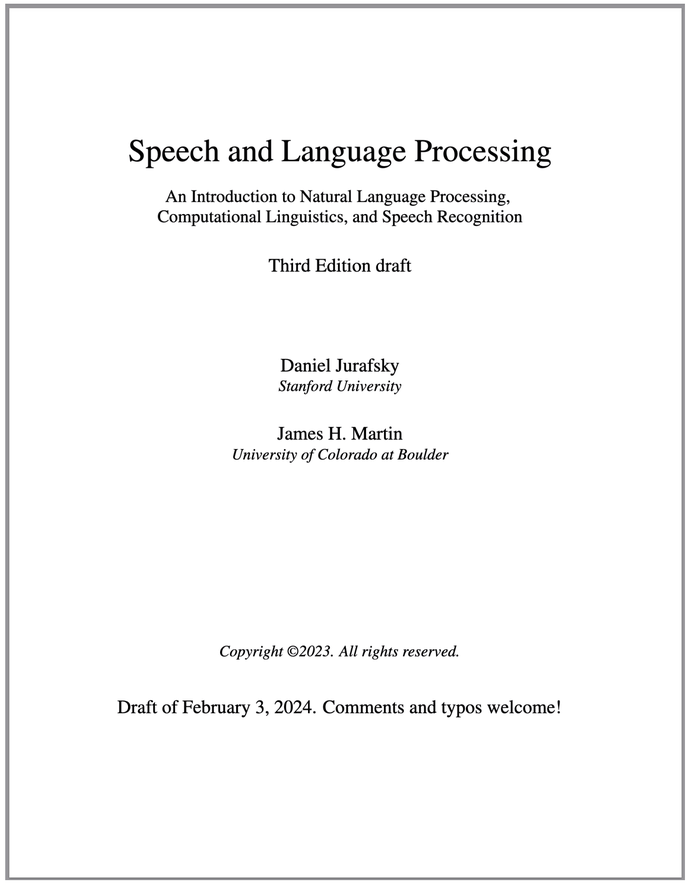
Summary of Contents

1 Introduction 1
I Words 19
2 Regular Expressions and Automata 21
3 Morphology and Finite-State Transducers 57
4 Computational Phonology and Text-to-Speech 91
5 Probabilistic Models of Pronunciation and Spelling 139
6 N-grams 189
7 HMMs and Speech Recognition 233
II Syntax 283
8 Word Classes and Part-of-Speech Tagging 285
9 Context-Free Grammars for English 319
10 Parsing with Context-Free Grammars 353
11 Features and Unification 391
12 Lexicalized and Probabilistic Parsing 443
13 Language and Complexity 473
III Semantics 495
14 Representing Meaning 497
15 Semantic Analysis 543
16 Lexical Semantics 587
17 Word Sense Disambiguation and Information Retrieval 627
IV Pragmatics 661
18 Discourse 663
19 Dialogue and Conversational Agents 715
20 Generation 759
21 Machine Translation 797
A Regular Expression Operators 829
B The Porter Stemming Algorithm 831
C C5 and C7 tagsets 835
D Training HMMs: The Forward-Backward Algorithm 841
Bibliography 851
Index 923



Source: Ross Girshick, the Parable of the Parser

NLP today has transform(er)ed



Speech and Language Processing

An Introduction to Natural Language Processing,
Computational Linguistics, and Speech Recognition

Third Edition draft

Daniel Jurafsky
Stanford University

James H. Martin
University of Colorado at Boulder

Copyright ©2023. All rights reserved.

Draft of February 3, 2024. Comments and typos welcome!

3rd edition

Summary of Contents

I Fundamental Algorithms for NLP 1
1 Introduction 3
2 Regular Expressions, Text Normalization, Edit Distance 4
3 N-gram Language Models 32
4 Naive Bayes, Text Classification, and Sentiment 60
5 Logistic Regression 81
6 Vector Semantics and Embeddings 105
7 Neural Networks and Neural Language Models 136
8 Sequence Labeling for Parts of Speech and Named Entities 162
9 RNNs and LSTMs 187
10 Transformers and Large Language Models 213
11 Fine-Tuning and Masked Language Models 242
12 Prompting, In-Context Learning, and Instruct Tuning 263

II NLP Applications 265
13 Machine Translation 267
14 Question Answering and Information Retrieval 293
15 Chatbots & Dialogue Systems 315
16 Automatic Speech Recognition and Text-to-Speech 337

III Annotating Linguistic Structure 365
17 Context-Free Grammars and Constituency Parsing 367
18 Dependency Parsing 391
19 Information Extraction: Relations, Events, and Time 415
20 Semantic Role Labeling 441
21 Lexicons for Sentiment, Affect, and Connotation 461
22 Coreference Resolution and Entity Linking 481
23 Discourse Coherence 511
Bibliography 533
Subject Index 563

The book is now 40% shorter

Neural networks, LMs, transformers, and LLMs

Misc. other stuff (including parsing)

Source: Ross Girshick, the Parable of the Parser

1999 → 2024: What happened to parsing?

"*In the early history of NLP these structures [parses] were an intermediate step toward deeper language processing.*

Source: Ross Girshick, the Parable of the Parser

1999 → 2024: What happened to parsing?

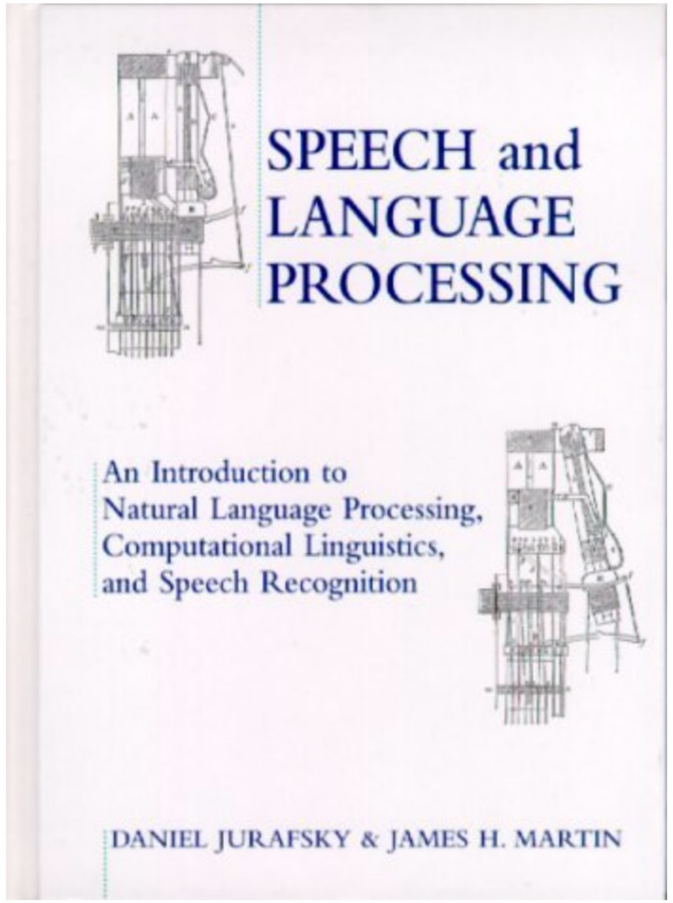
"*In the early history of NLP these structures [parses] were an intermediate step toward deeper language processing.*

In modern NLP, we don't generally make explicit use of parse or other structures inside the neural language models [...], or directly in applications like those we discussed in Part II.

- Jurafsky & Martin

Source: Ross Girshick, the Parable of the Parser

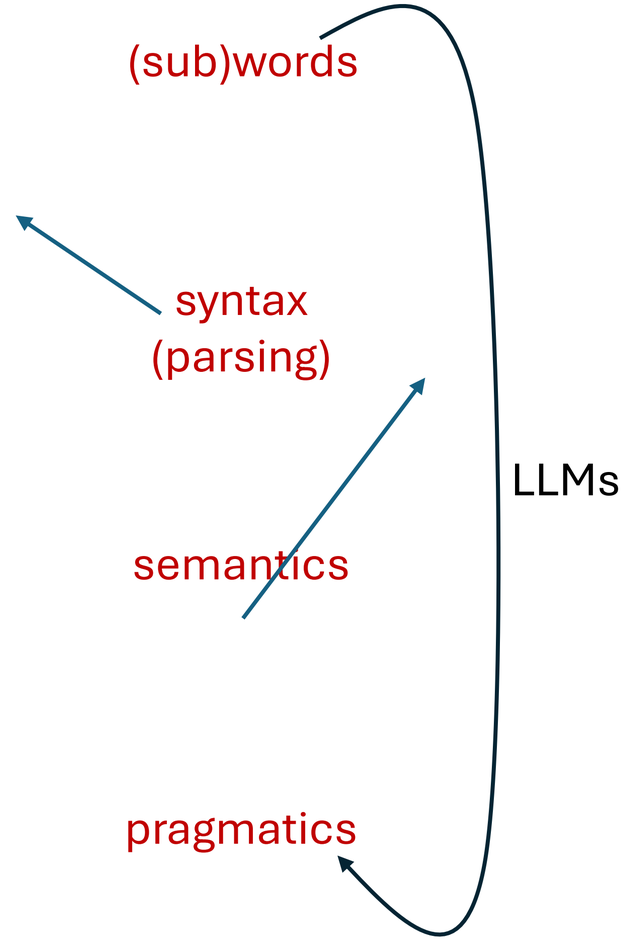
NLP circa 1999



1st edition

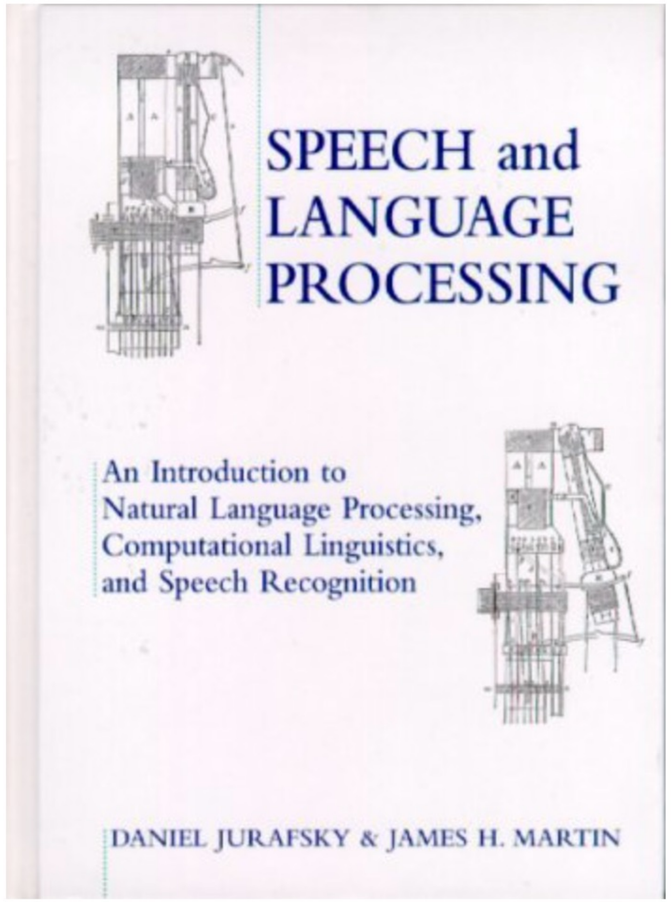
Summary of Contents

1 Introduction 1
I Words 19
2 Regular Expressions and Automata 21
3 Morphology and Finite-State Transducers 57
4 Computational Phonology and Text-to-Speech 91
5 Probabilistic Models of Pronunciation and Spelling 139
6 N-grams 189
7 HMMs and Speech Recognition 233
II Syntax 283
8 Word Classes and Part-of-Speech Tagging 285
9 Context-Free Grammars for English 319
10 Parsing with Context-Free Grammars 353
11 Features and Unification 391
12 Lexicalized and Probabilistic Parsing 443
13 Language and Complexity 473
III Semantics 495
14 Representing Meaning 497
15 Semantic Analysis 543
16 Lexical Semantics 587
17 Word Sense Disambiguation and Information Retrieval 627
IV Pragmatics 661
18 Discourse 663
19 Dialogue and Conversational Agents 715
20 Generation 759
21 Machine Translation 797
A Regular Expression Operators 829
B The Porter Stemming Algorithm 831
C C5 and C7 tagsets 835
D Training HMMs: The Forward-Backward Algorithm 841
Bibliography 851
Index 923



Source: Ross Girshick, the Parable of the Parser

NLP circa 1999



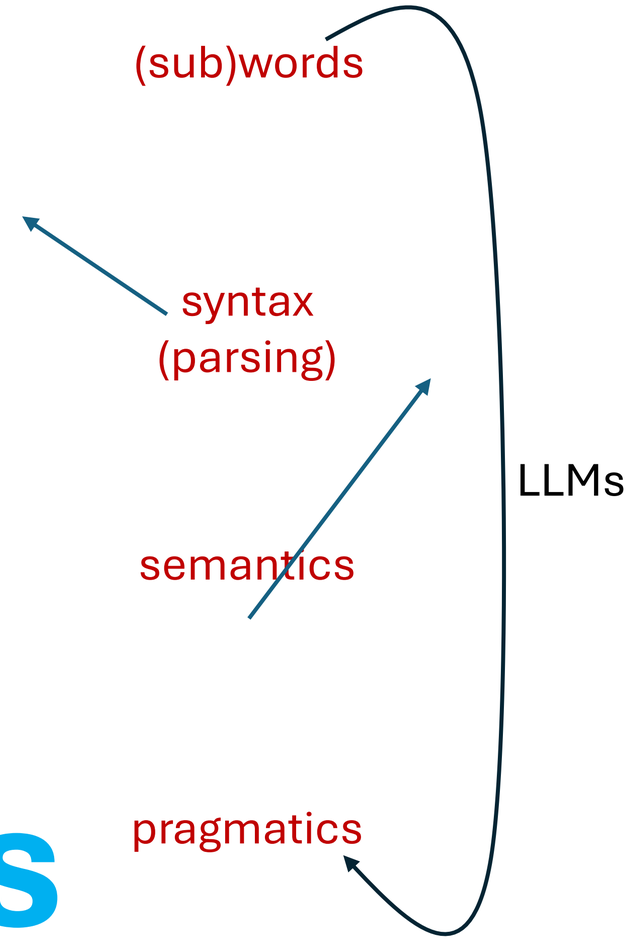
1st edition

Summary of Contents

1 Introduction 1
I Words 19
2 Regular Expressions and Automata 21
3 Morphology and Finite-State Transducers 57
4 Computational Phonology and Text-to-Speech 91
5 Probabilistic Models of Pronunciation and Spelling 139
6 N-grams 189
7 HMMs and Speech Recognition 233
II Syntax 283
8 Word Classes and Part-of-Speech Tagging 285
9 Context-Free Grammars for English 319
10 Parsing with Context-Free Grammars 353
11 Features and Unification 391
12 Lexicalized and Probabilistic Parsing 443
13 Language and Complexity 473
III Semantics 495
14 Representing Meaning 497
15 Semantic Analysis 543
16 Lexical Semantics 587
17 Word Sense Disambiguation and Information Retrieval 627
IV Pragmatics 661
18 Discourse 663
19 Dialogue and Conversational Agents 715
20 Generation 759
21 Machine Translation 797
A Regular Expression Operators 829
B The Porter Stemming Algorithm 1
C C5 and C7 Tagsets 5
D Training HMMs: The Forward-Backward Algorithm 1
Bibliography 1
Index 923

Fake Tasks

Real Tasks



Source: Ross Girshick, the Parable of the Parser

If your end goals are chat, MT, etc., then do not build a parser.

(Of course, if your end goal is to have a parser, then build a parser. But likely you will do this with an LLM anyway.)



Source: Ross Girshick, the Parable of the Parser

What are the “parsers” of computer vision?

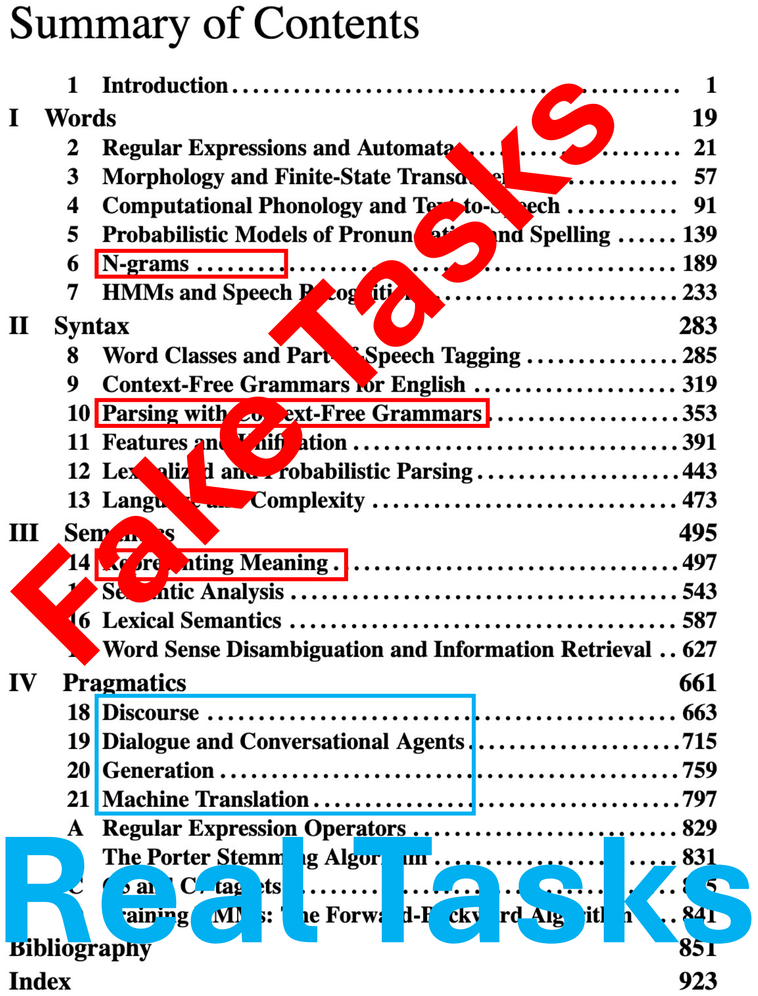
- Before we answer this question, take a step back
- We need to understand what our real tasks are
- Let’s study the rise of LLMs
  - What lessons can we learn?



Source: Ross Girshick, the Parable of the Parser

Why are LLMs so successful?

- *Real* tasks are text generation tasks
  - High societal and economic value
  - Universality: can do POS tagging, parsing, etc. *as text generation* (if you *really* want to)



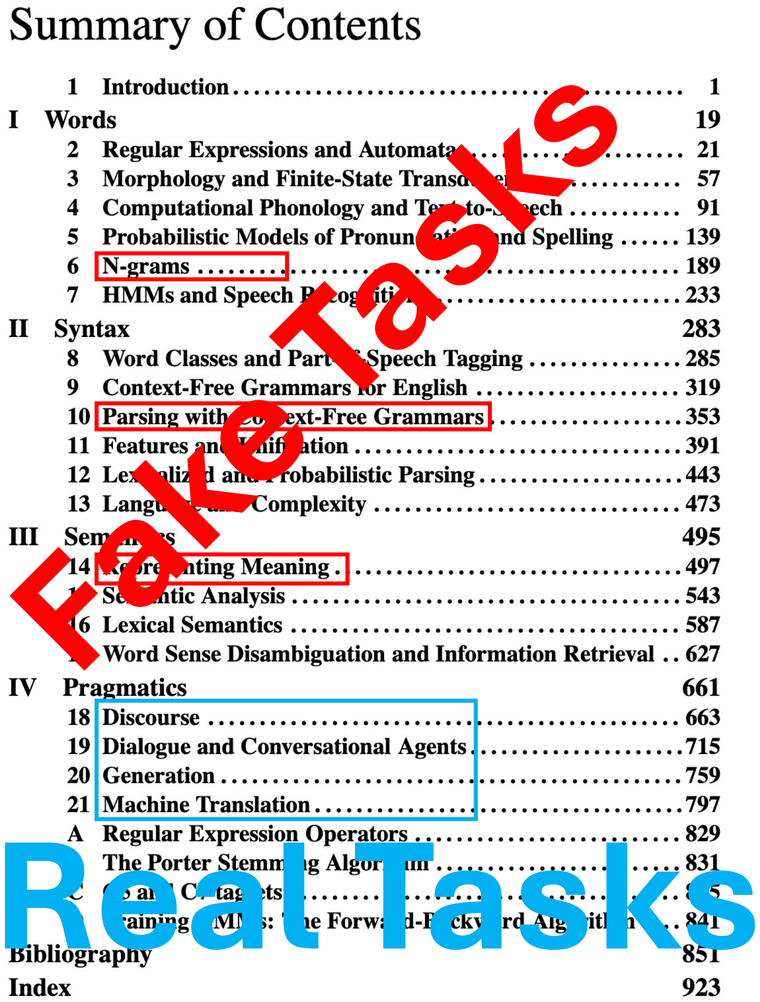
NLP circa 2024

NLP circa 1999

Source: Ross Girshick, the Parable of the Parser

Why are LLMs so successful?

- *Real* tasks are text generation tasks
  - High societal and economic value
  - Universality: can do POS tagging, parsing, etc. *as text generation* (if you *really* want to)
- **Text generation is all you need**



NLP circa 2024

NLP circa 1999

Source: Ross Girshick, the Parable of the Parser

Why are LLMs so successful?

- *Real* tasks are text generation tasks
  - High societal and economic value
  - Universality: can do POS tagging, parsing, etc. *as text generation* (if you *really* want to)
- **Text generation is all you need**
- **But wait, there's more!**
  What if I told you:
  - *High-quality data is abundant*
  - *The test task = the training task*
  - *Training is basically supervised*

NLP circa 2024

Summary of Contents

1 Introduction 1
I Words 19
2 Regular Expressions and Automata 21
3 Morphology and Finite-State Transducers 57
4 Computational Phonology and Text-to-Speech 91
5 Probabilistic Models of Pronunciation and Spelling 139
6 N-grams .....
7 HMMs and Speech Recognition .....
II Syntax
8 Word Classes and Part-of-Speech Tagging ..
9 Context-Free Grammars for English .....
10 Parsing with Context-Free Grammars .....
11 Features and Unification .....
12 Lexicalized and Probabilistic Parsing .....
13 Language and Complexity .....
III Semantics
14 Representing Meaning .....
15 Semantic Analysis .....
16 Lexical Semantics .....
Word Sense Disambiguation and Information
IV Pragmatics
18 Discourse .....
19 Dialogue and Conversational Agents .....
20 Generation .....
21 Machine Translation .....
A Regular Expression Operators 829
The Porter Stemming Algorithm 831
C and C tagsets .....
Training HMMs: The Forward-Backward Algorithm 841
Bibliography 851
Index 923

Fake Tasks



Real Tasks

NLP circa 1999

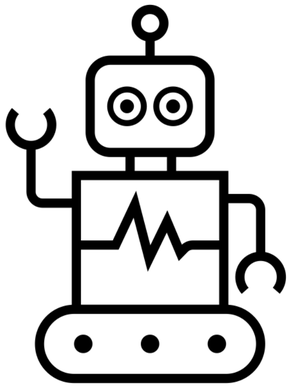
Source: Ross Girshick, the Parable of the Parser

What are the real tasks of computer vision?

- Caveat: focusing on recognition-y things, not 3D-y things

- Q1: Who benefits from CV?
- A1: Those who cannot see
  - People with low or no vision
    - Real task = open-ended visual QA about the real world
  - In general, AI agents operating in visual environments
    - Real tasks =
      - Robots: “cook me dinner”, “fold my laundry”, “wash my dishes”, …
      - Internet agents: “shop for my climbing shoes at a good sale price”, …
      - …

VizWiz



Source: Ross Girshick, the Parable of the Parser

Vincent's (Current) Hot Take

- A major chunk of vision problems (3D reconstruction, segmentation, classification, tracking…) were motivated by robotics
- But robotic's input-output behavior is: video in, actions out
  - Hence, as a corollary to the bitter lesson, 3D and other "intermediate representations" may be unnecessary
- Another "real task" in vision is question answering: helping humans disseminate videos quickly in language
  - VLMs have "solved" this for natural language
  - Similarly, no 3D reconstruction, segmentation, classification etc were necessary here

- Further reading:
  - My blog post: https://www.vincentsitzmann.com/blog/bitter_lesson_of_cv/
  - Ross Girshick's "Parable of the Parser": https://drive.google.com/file/d/1VodGIjuEhBKwZIXQwN-ApH6g2wBAVAdK/view

In this context: Video Generative Models





Does it make sense to ask “Do these models *understand* 3D” or “How can we use these for segmentation”? Does it *matter?* What are the “real tasks” in vision?

In this context: Video Generative Models





Does it make sense to ask “Do these models *understand* 3D” or “How can we use these for segmentation”? Does it *matter?* What are the “real tasks” in vision?

In this context: Video Generative Models

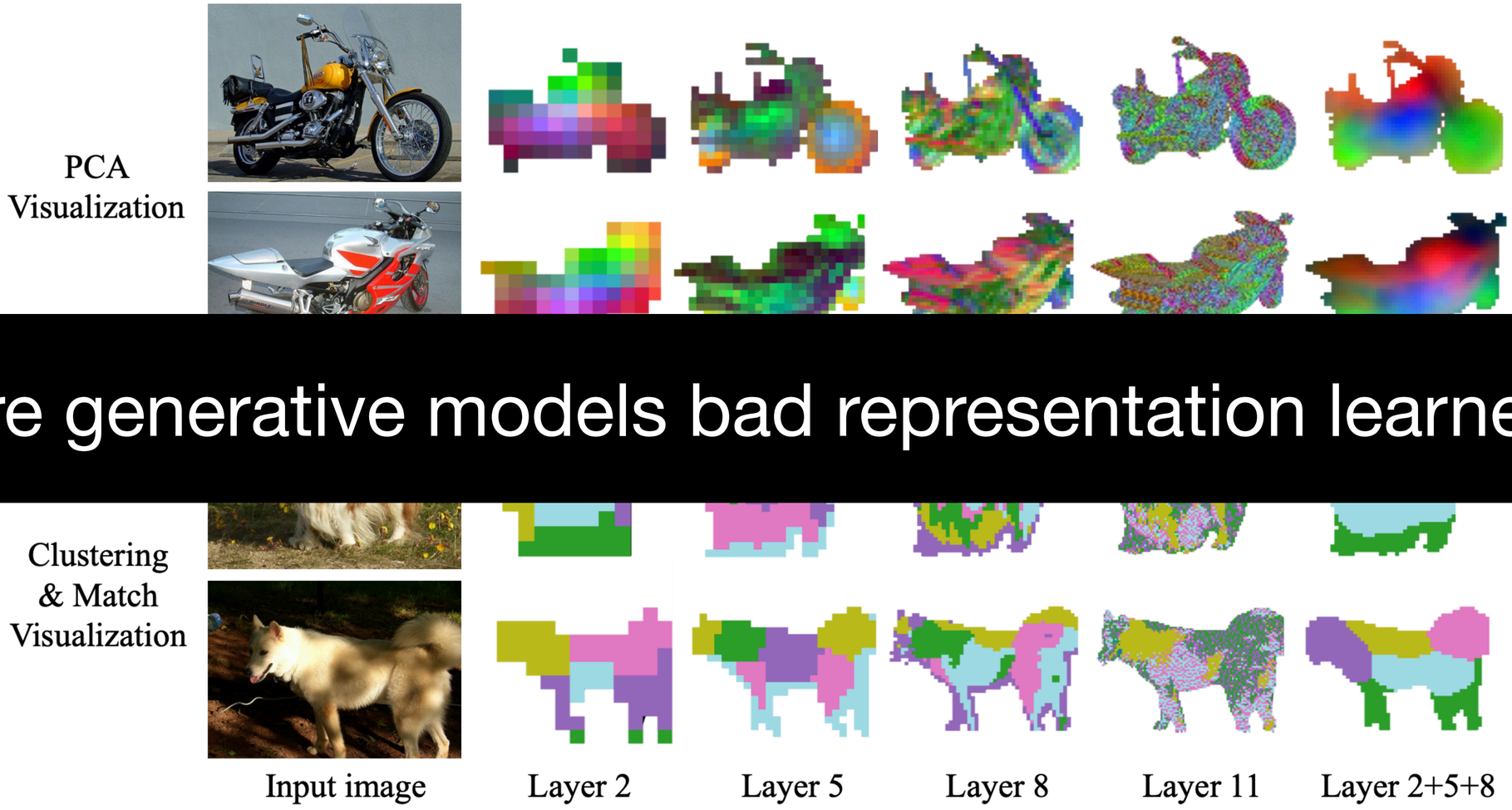




Does it make sense to ask “Do these models *understand* 3D” or “How can we use these for segmentation”? Does it *matter?* What are the “real tasks” in vision?

Sources: Runway, OpenAI

While latent representations of generative models have *some* info, they fall far behind of dedicated SSL systems



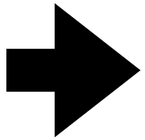
Are generative models bad representation learners?

Figure 2: **Analysis of features from different decoder layers in SD.**

Zhang et al., A Tale of Two Features: Stable Diffusion Complements DINO for Zero-Shot Semantic Correspondence, 2023

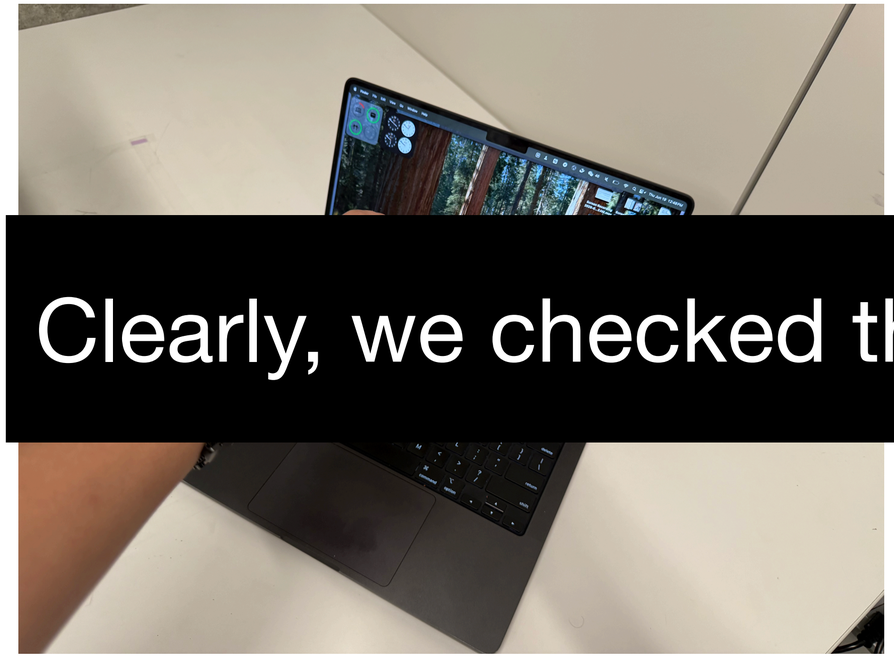
The Large Video Planner (Chen et al. 2025)





“Close the Laptop”

The Large Video Planner (Chen et al. 2025)

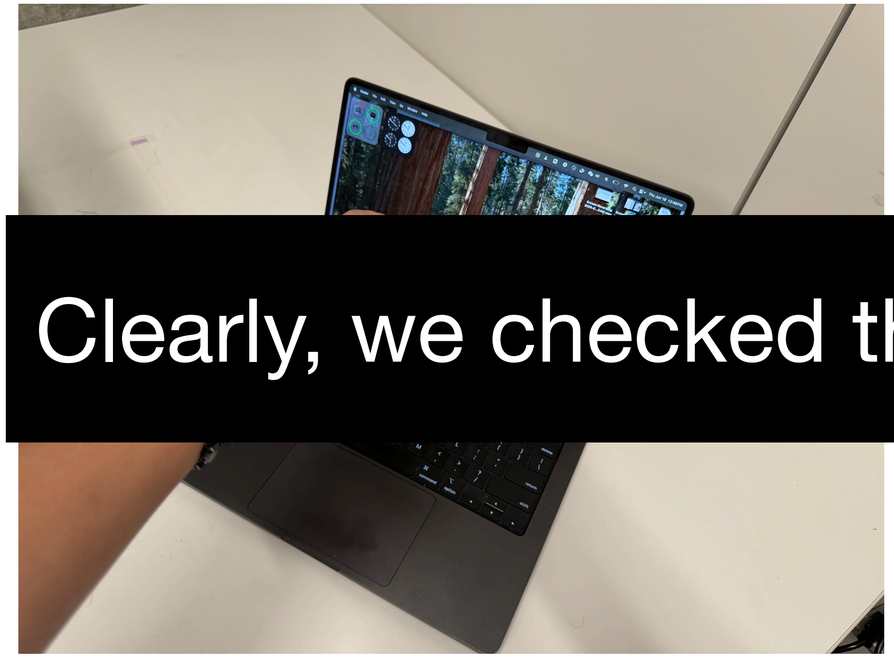


“Close the Laptop”



Clearly, we checked the box of “Useful to Downstream Task”!

The Large Video Planner (Chen et al. 2025)

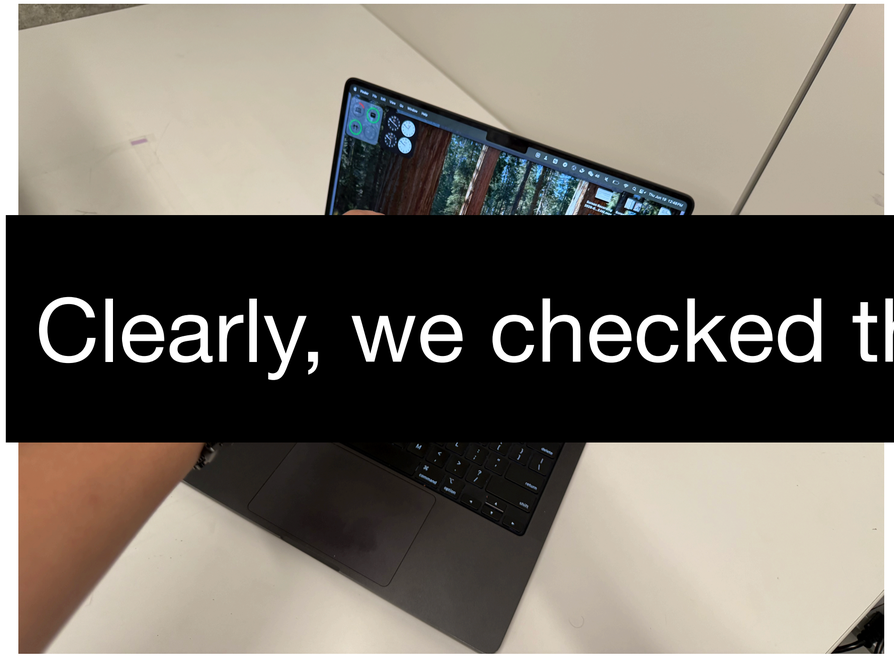


“Close the Laptop”



Clearly, we checked the box of “Useful to Downstream Task”!

The Large Video Planner (Chen et al. 2025)



“Close the Laptop”



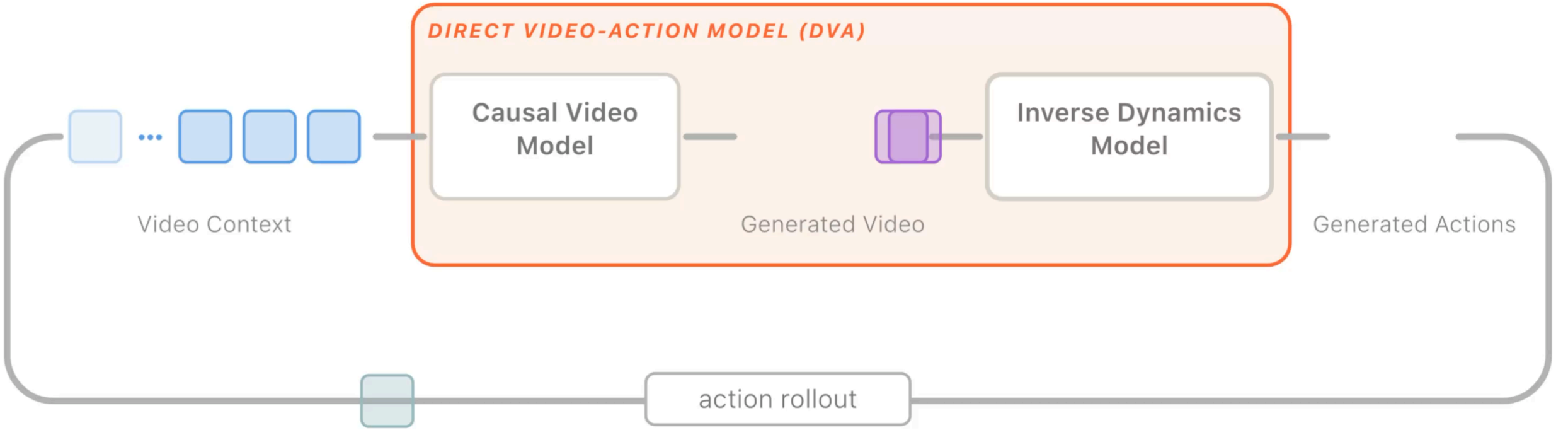
Clearly, we checked the box of “Useful to Downstream Task”!



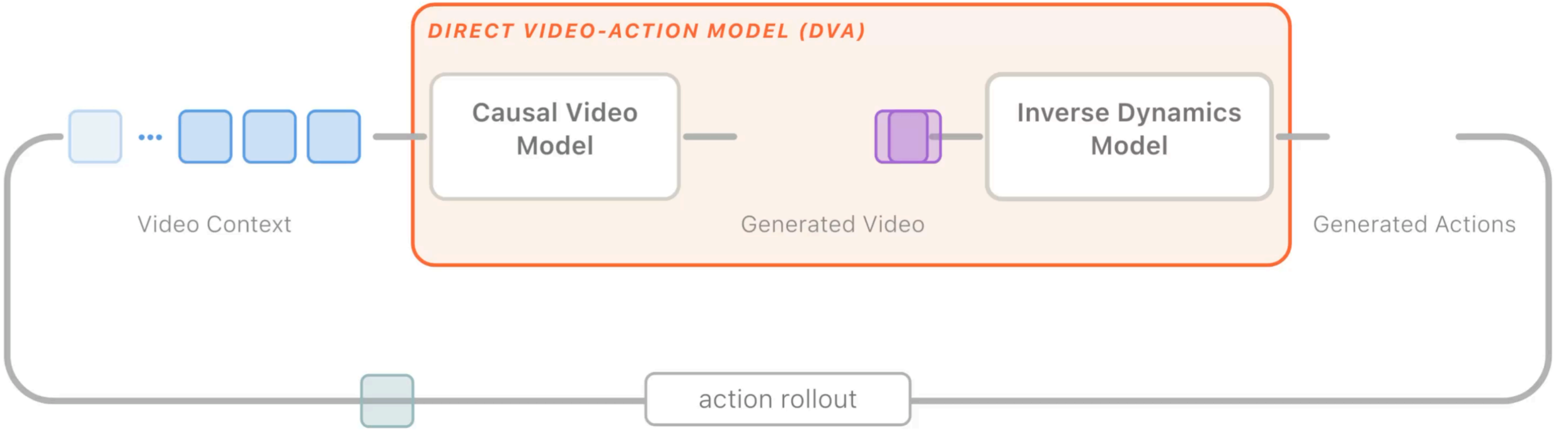
Disclaimer: in discussion to advise company!



Disclaimer: in discussion to advise company!



Source: Rhoda AI, https://www.rhoda.ai/research/direct-video-action



Source: Rhoda AI, https://www.rhoda.ai/research/direct-video-action

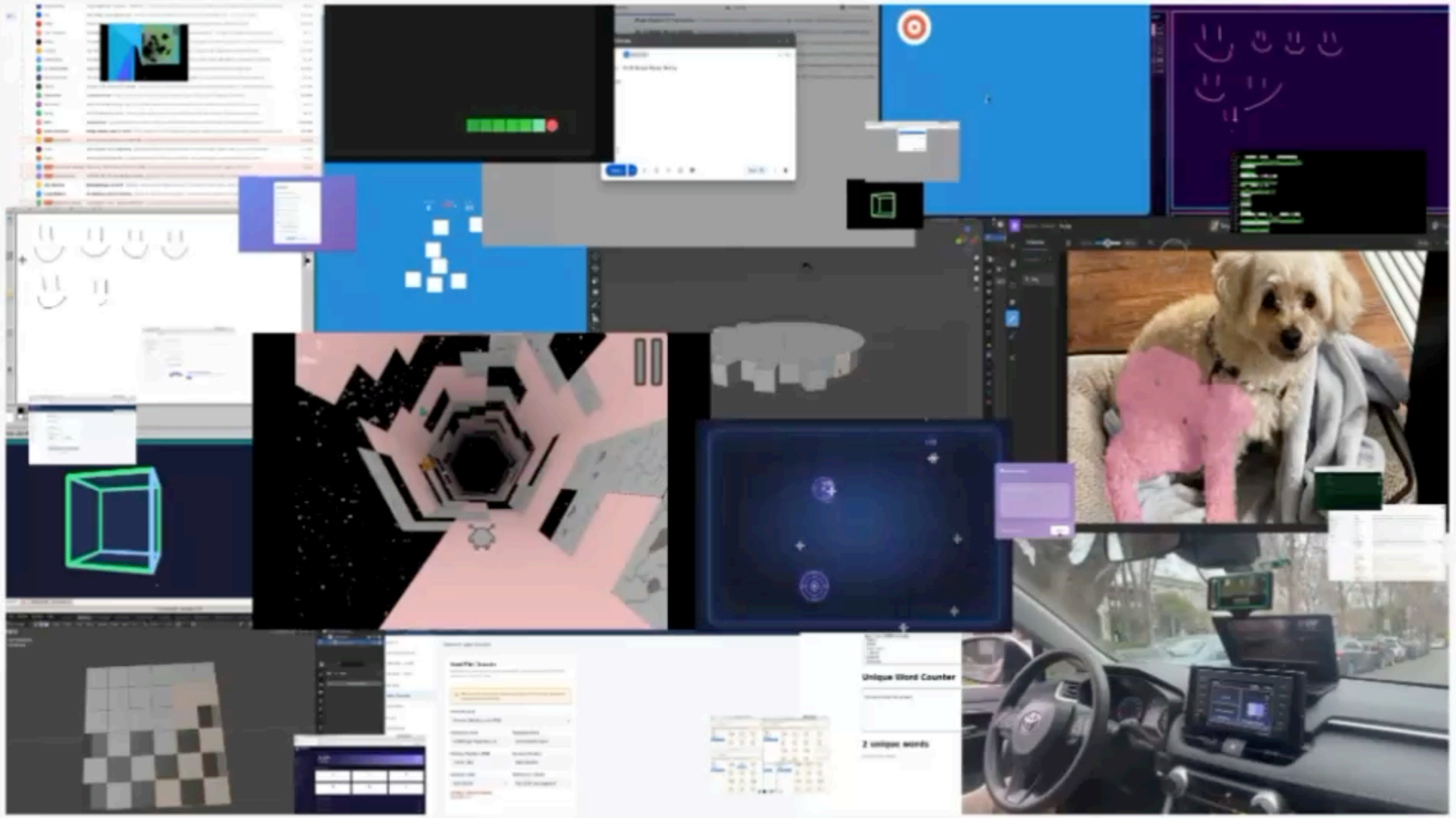
Similar idea can control computers: learn to generate videos of tasks, then back out mouse & keyboard commands!

Source: Standard Intelligence

The First Fully General Computer Action Model

We trained a model on our 11-million-hour video dataset. Our model can explore complex websites, complete multi-action CAD modeling sequences, and drive a car in the real world, all at 30 FPS.

Standard Intelligence Team February 23, 2026



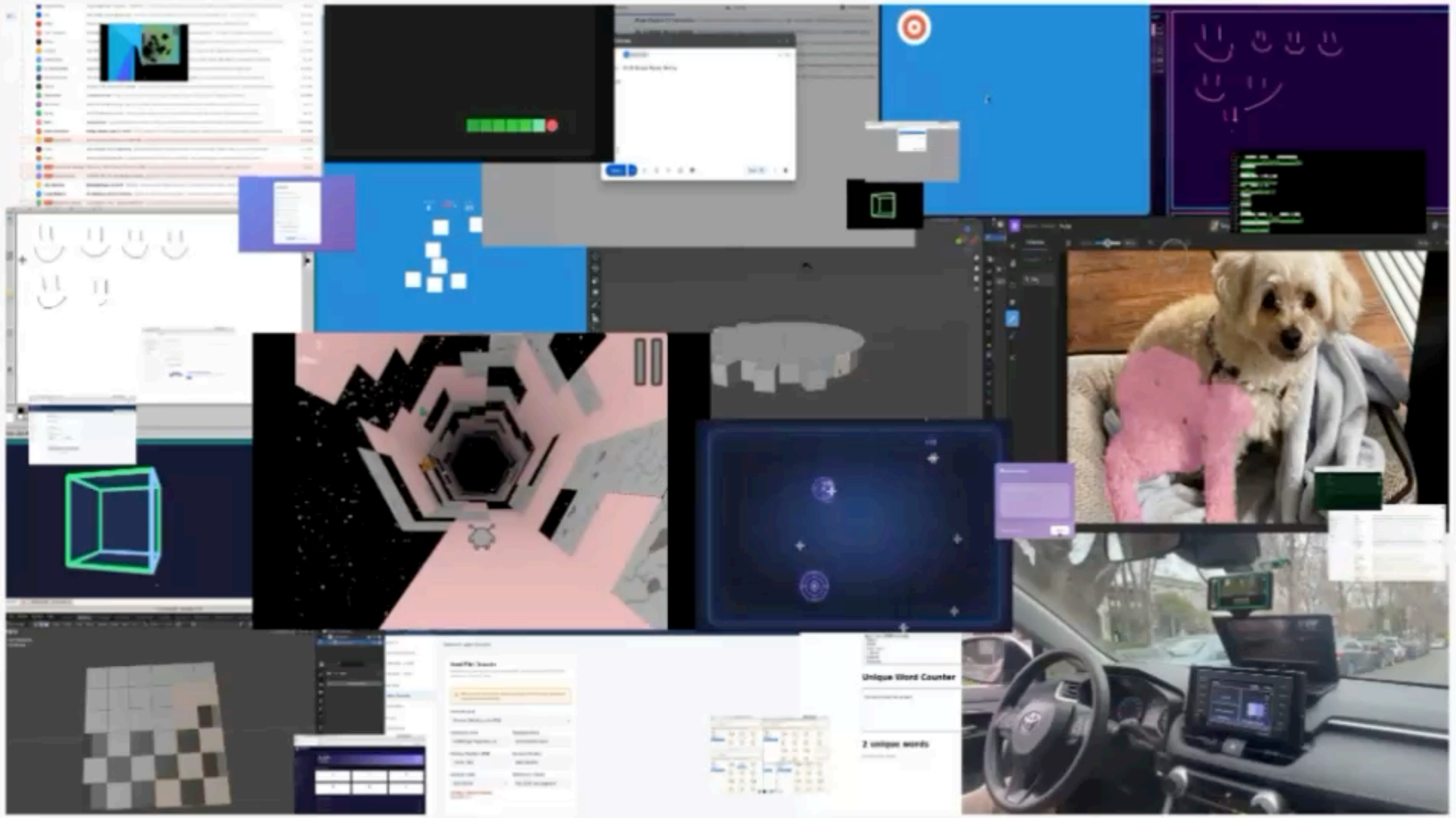
Similar idea can control computers: learn to generate videos of tasks, then back out mouse & keyboard commands!

Source: Standard Intelligence

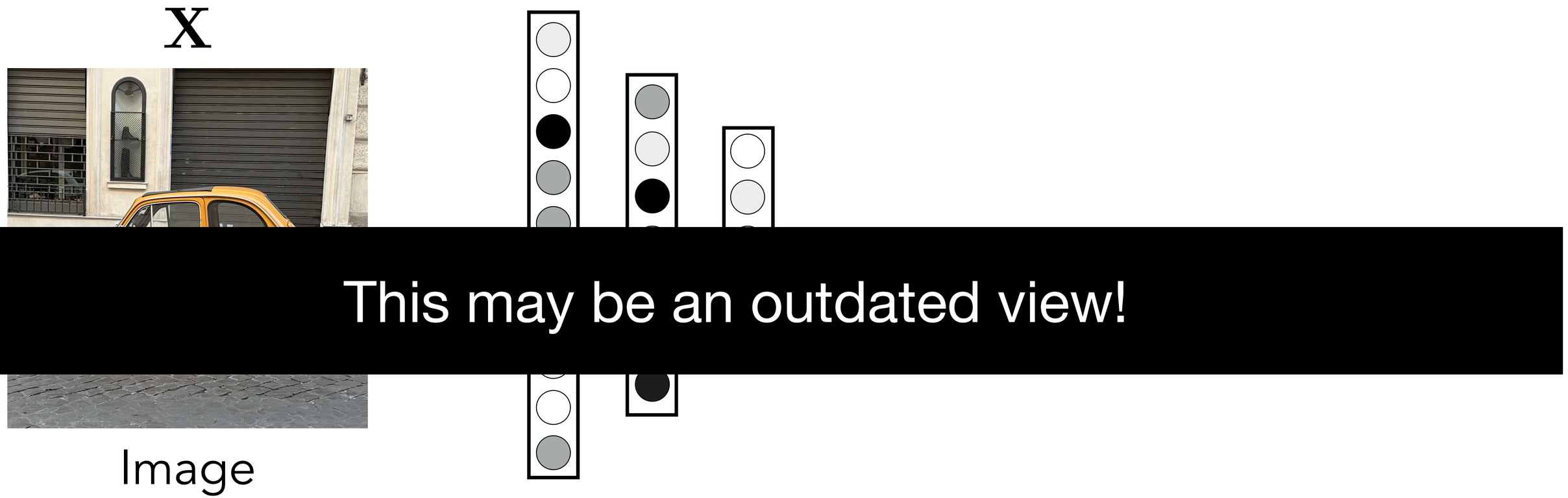
The First Fully General Computer Action Model

We trained a model on our 11-million-hour video dataset. Our model can explore complex websites, complete multi-action CAD modeling sequences, and drive a car in the real world, all at 30 FPS.

Standard Intelligence Team February 23, 2026

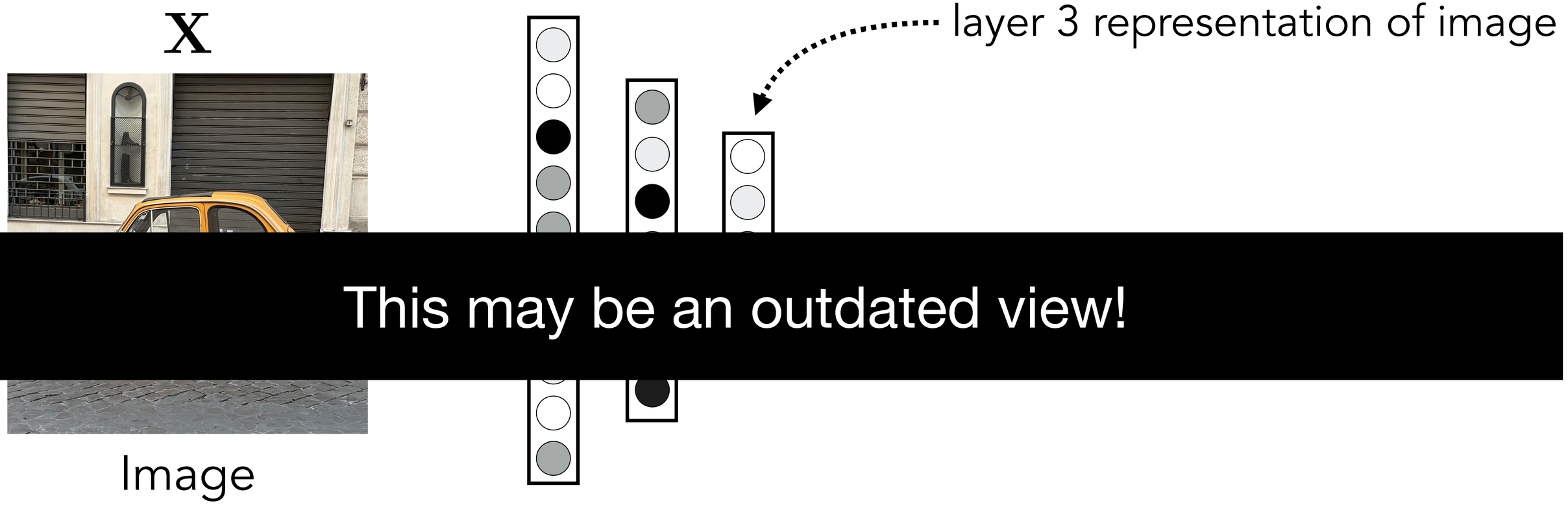


Conclusion: Representation Learning = Pre-training a model for later fine-tuning



Given a signal, extract a *latent representation* of it that is useful to downstream tasks.

Conclusion: Representation Learning = Pre-training a model for later fine-tuning



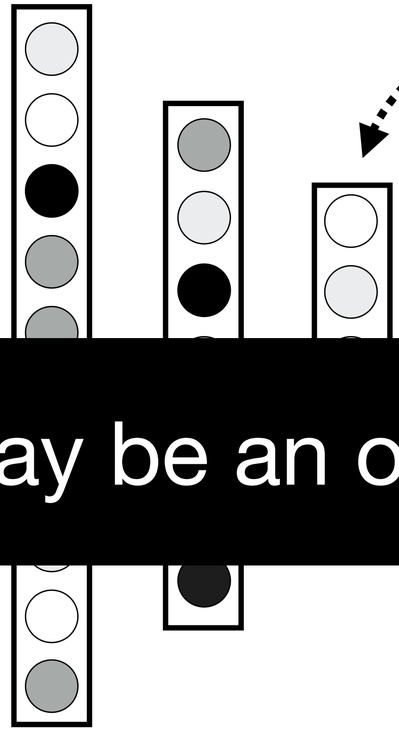
Given a signal, extract a *latent representation* of it that is useful to downstream tasks.

Conclusion: Representation Learning = Pre-training a model for later fine-tuning

$\mathbf{x}$



Image



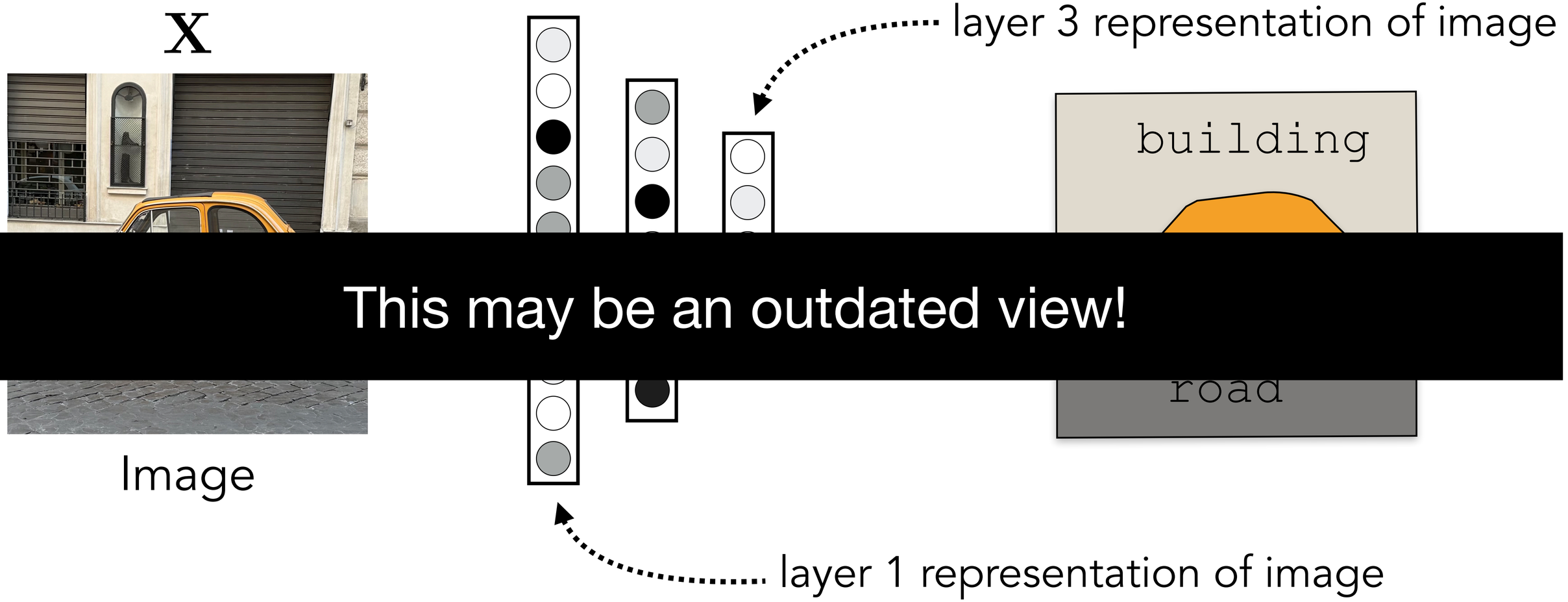
layer 3 representation of image

layer 1 representation of image

This may be an outdated view!

Given a signal, extract a *latent representation* of it that is useful to downstream tasks.

Conclusion: Representation Learning = Pre-training a model for later fine-tuning



This may be an outdated view!

Given a signal, extract a *latent representation* of it that is useful to downstream tasks.

Conclusion: Representation Learning = Pre-training a model for later fine-tuning



Imag

layer 3 rep tion of image

building

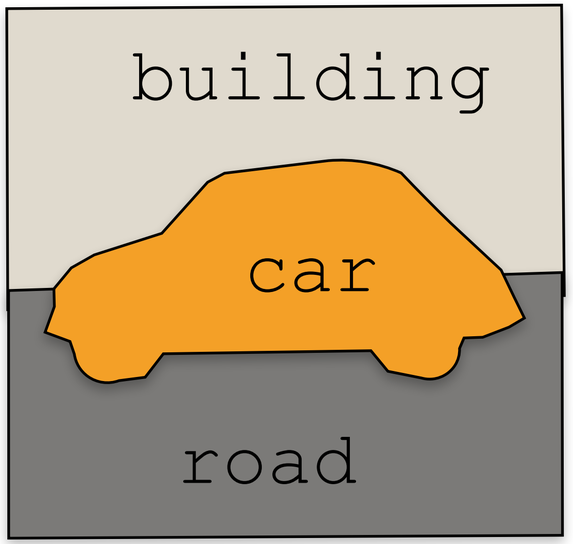
road

layer 1 representation of image

This may be an outdated view!

Given a signal, extract a *latent representation* of it that is useful to downstream tasks.

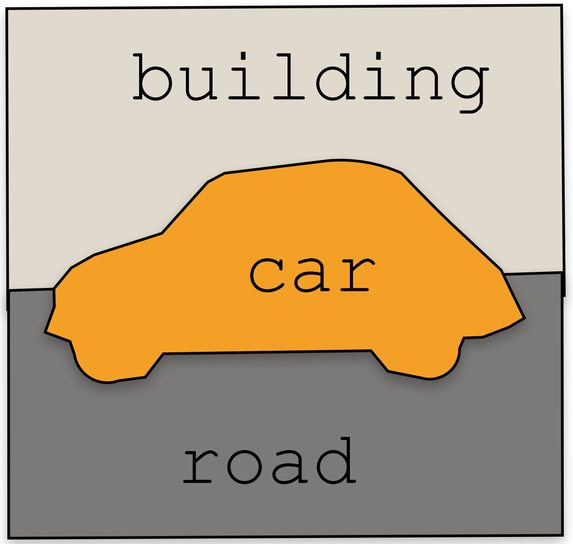
What Makes a Good Representation?



[See “Representation Learning”, Bengio 2013, for more commentary]

What Makes a Good Representation?

Good representations are:

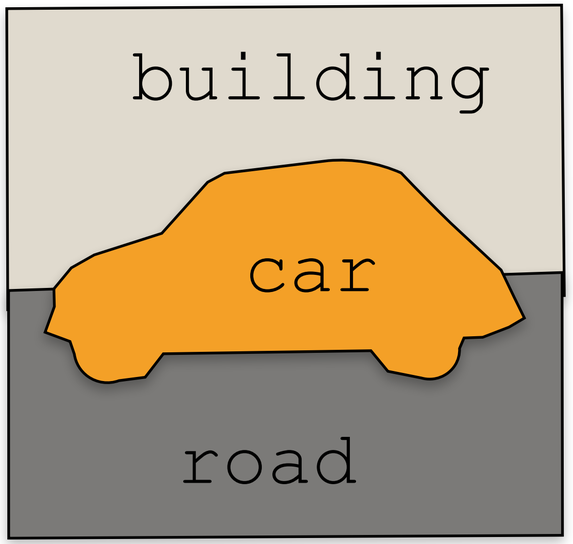


[See “Representation Learning”, Bengio 2013, for more commentary]

What Makes a Good Representation?

Good representations are:

1. Compact (*minimal*)

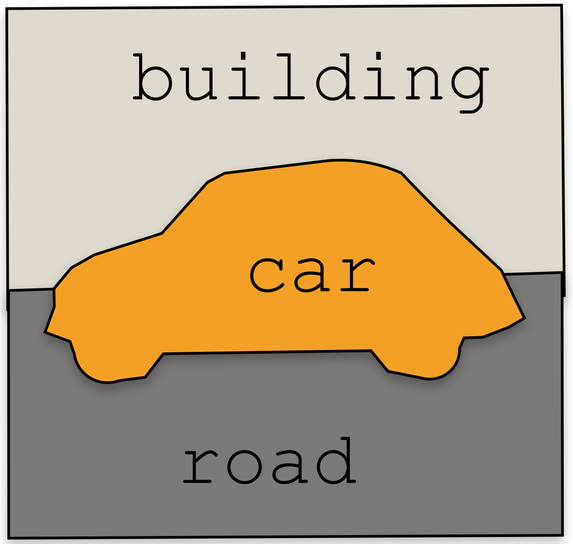


[See “Representation Learning”, Bengio 2013, for more commentary]

What Makes a Good Representation?

Good representations are:

1. Compact (*minimal*)
2. Explanatory (*sufficient*)

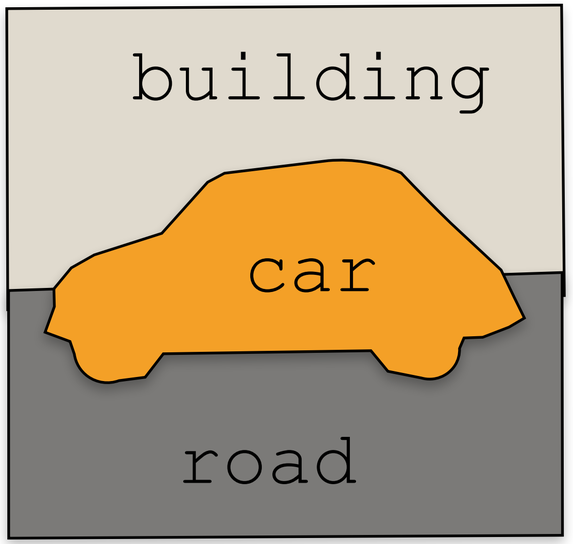


[See “Representation Learning”, Bengio 2013, for more commentary]

What Makes a Good Representation?

Good representations are:

1. Compact (*minimal*)
2. Explanatory (*sufficient*)
3. Disentangled (*independent factors*)

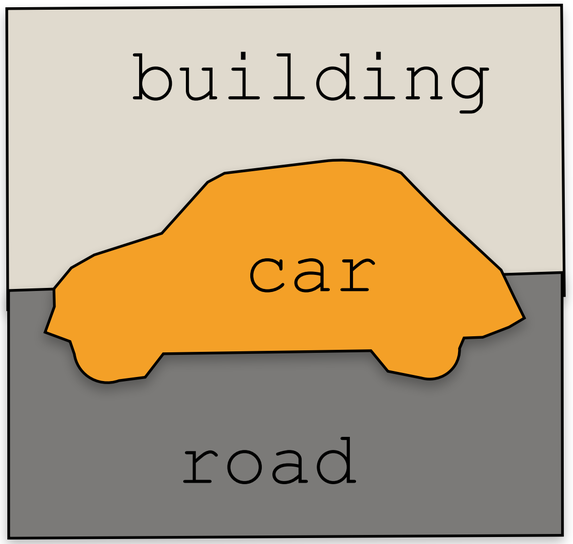


[See “Representation Learning”, Bengio 2013, for more commentary]

What Makes a Good Representation?

Good representations are:

1. Compact (*minimal*)
2. Explanatory (*sufficient*)
3. Disentangled (*independent factors*)
4. Equivariant (*transform predictably*)

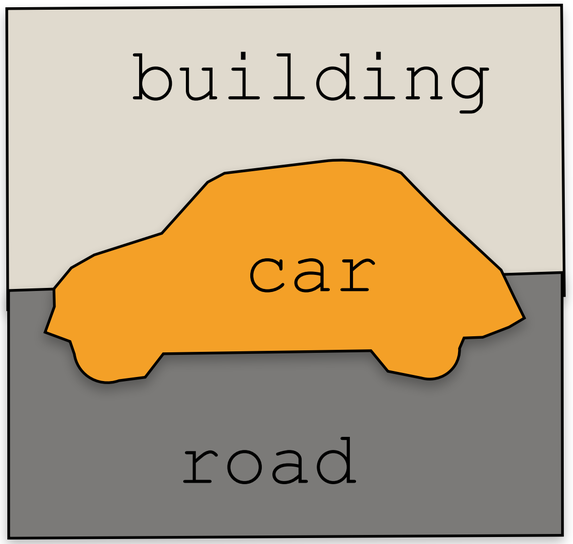


[See “Representation Learning”, Bengio 2013, for more commentary]

What Makes a Good Representation?

Good representations are:

1. Compact (*minimal*)
2. Explanatory (*sufficient*)
3. Disentangled (*independent factors*)
4. Equivariant (*transform predictably*)
5. Interpretable

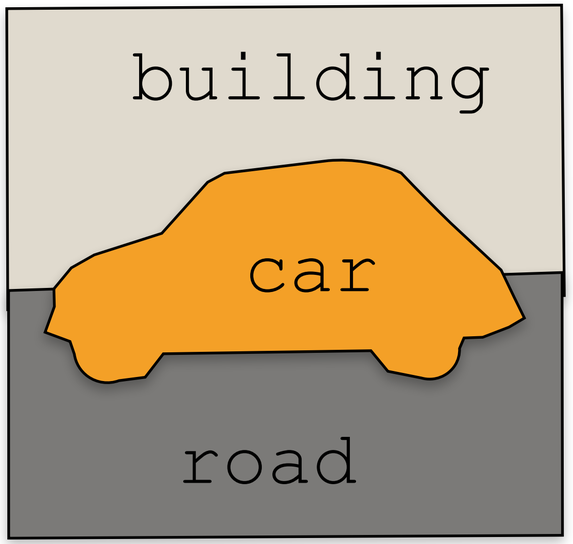


[See “Representation Learning”, Bengio 2013, for more commentary]

What Makes a Good Representation?

Good representations are:

1. Compact (*minimal*)
2. Explanatory (*sufficient*)
3. Disentangled (*independent factors*)
4. Equivariant (*transform predictably*)
5. Interpretable
6. *Make subsequent problem solving easy*

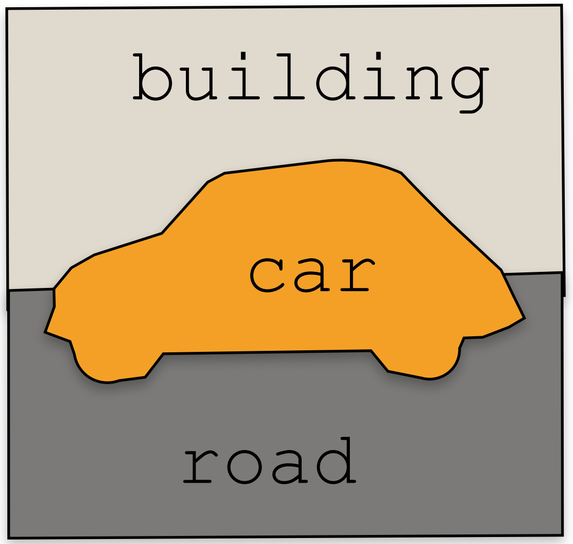


[See “Representation Learning”, Bengio 2013, for more commentary]

What Makes a Good Representation?

Good representations are:

1. Compact (*minimal*)
2. Explanatory (*sufficient*)
3. Disentangled (*independent factors*)
4. Equivariant (*transform predictably*)
5. Interpretable
6. *Make subsequent problem solving easy*
7. …?



[See “Representation Learning”, Bengio 2013, for more commentary]

What Makes a Good Representation?

Good representations are:

1. Compact (*minimal*)
2. Explanatory (*sufficient*)
3. Disentangled (*independent factors*)
4. Equivariant (*transform predictably*)
5. Interpretable

Perhaps these **do not matter…**

…and only this one matters.

6. *Make subsequent problem solving easy*

7. …?

[See “Representation Learning”, Bengio 2013, for more commentary]

The rest of this module

- Two lectures by Peter Holderrieth on the fundamentals of flow matching and diffusion
- Then three lectures by Vincent going deeper on image diffusion, as well as sequence generation
- Then one lecture on self-supervised learning and latent world models
- Finally a guest lecture by Mathilde Caron on her work on image-based SSL