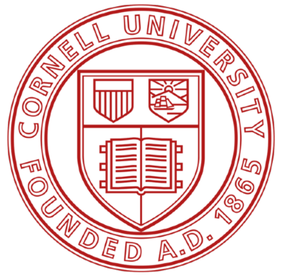
Lecture 17: Adapting generative models to other tasks

CS 5788: Introduction to Generative Models



Announcements

- PS1 grades coming soon

We have generative models that can generate high quality language, images, audio.

How do we use these models to solve the “downstream” tasks that we actually care about?

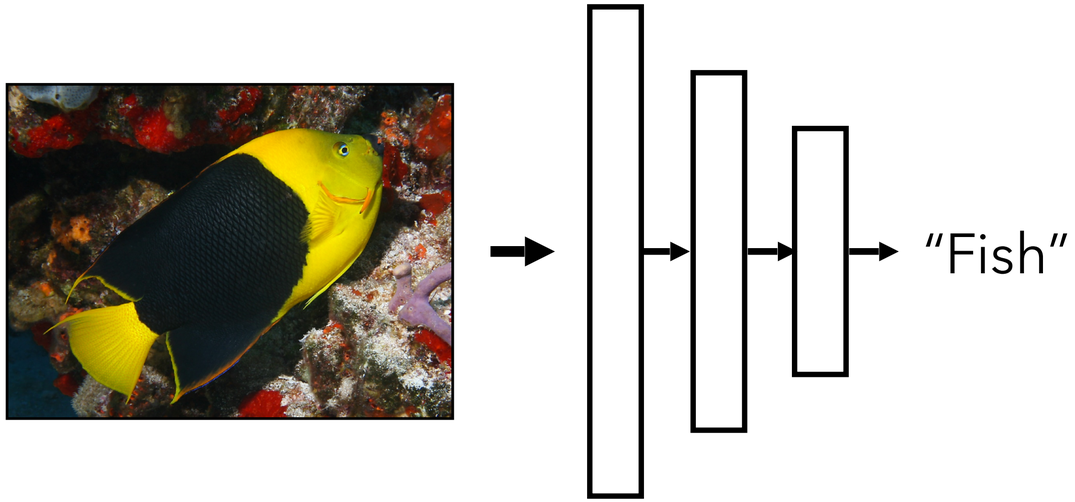
Today

- Adapting a network to new tasks
- Generative pretraining
- Zero-shot adaptation

Transfer learning

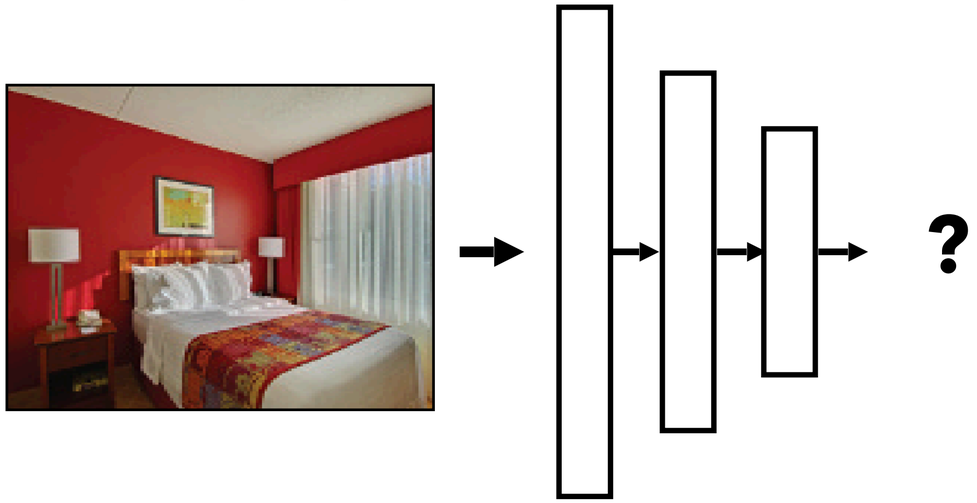
Training

Recognizing objects, generating sentences, etc.



Testing

Related task: Recognizing scenes, classifying documents, etc.

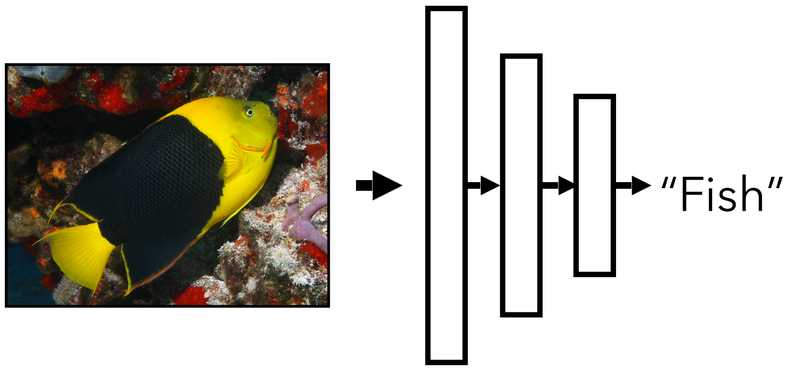


Often, what we will be "tested" on is to learn to do something new.

Source: Isola, Freeman, Torralba

Pretraining

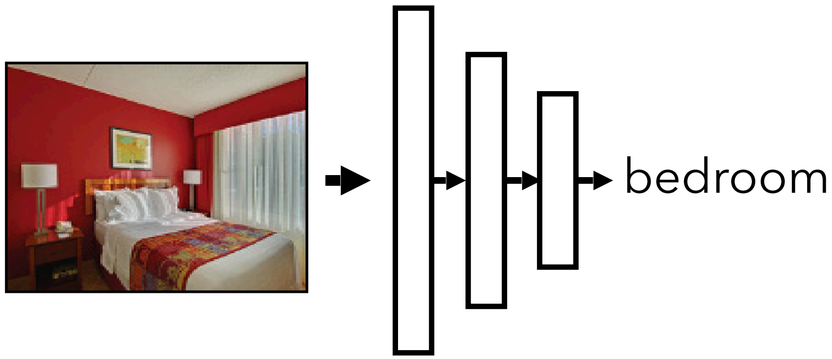
e.g., Object recognition



A lot of data

Finetuning

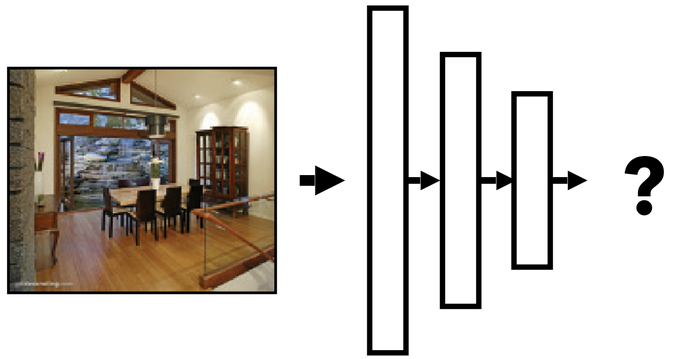
e.g., Scene recognition



A little data

Testing

e.g., Scene recognition



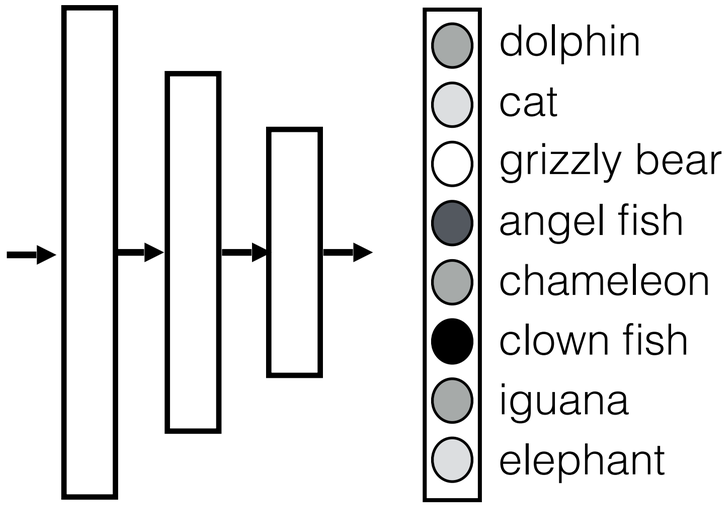
Finetuning: take a model trained on one task and retrain it for another.

Source: Isola, Freeman, Torralba

Finetuning

Pretraining

Object recognition



Finetuning

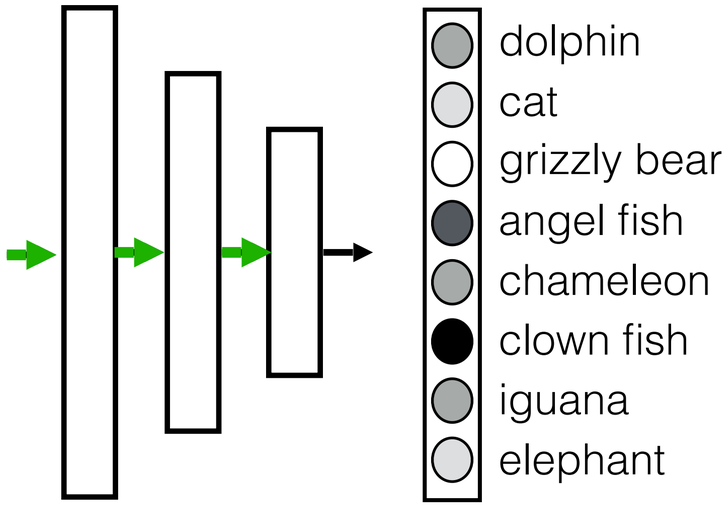
Scene recognition

Source: Isola, Freeman, Torralba

Finetuning

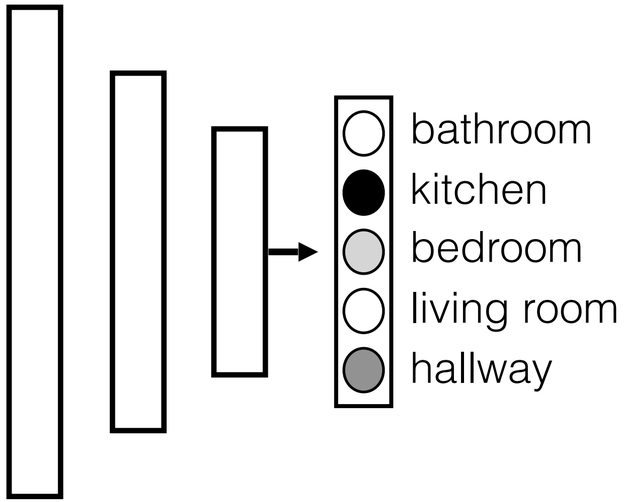
Pretraining

Object recognition



Finetuning

Scene recognition



Initialize the weights using the pretraining task!

Source: Isola, Freeman, Torralba

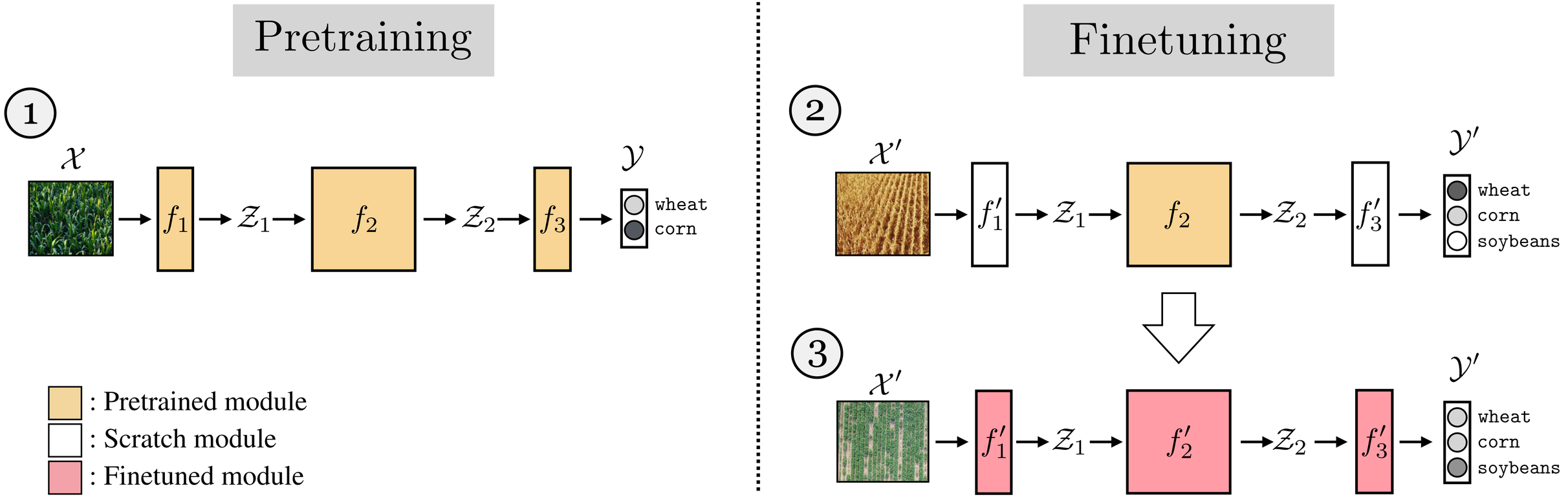
Finetuning

- Pretrain a network on task A, resulting in parameters Φ.
- Initialize a second network with some or all of Φ.
- Train the second network on task B, resulting in parameters Φ'.
- We can think of this as a small update to the original weights, e.g., $\Phi' = \Phi + \Delta\Phi$.

Adapted from Isola, Freeman, Torralba

Architectures during finetuning

What if the input/output dimensionality needs to change?
Can introduce layers (typically later) initialized from scratch.



Source: Isola, Freeman, Torralba. FCV book.

How do you avoid forgetting the pretraining?

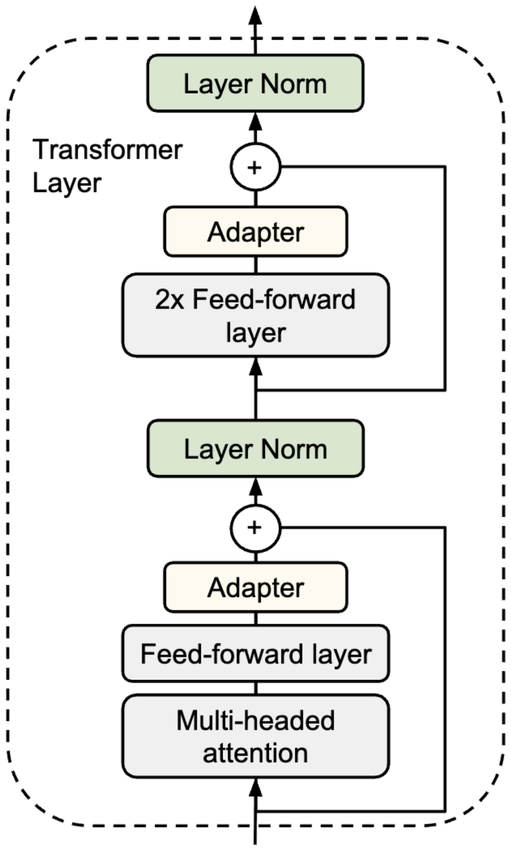
- Avoiding overfitting on a small amount of data for a new task while still enabling specialization can be a balancing act. A few methods:
  - Freeze most of the network, train only the final layers
  - Use a small learning rate
  - Early stopping
  - Continue to train on the original data as well
- Often lots of trial and error.
- Very helpful to have a good, representative validation set for your task.

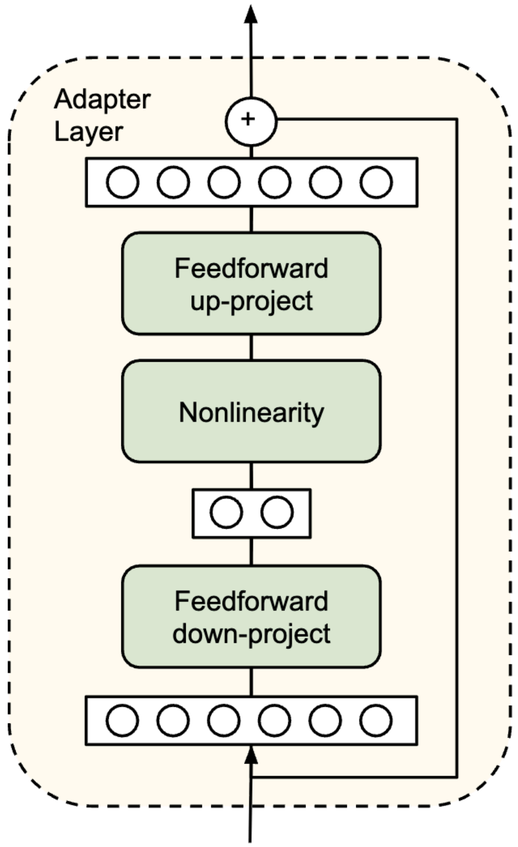
Adapted from P. Isola and S. Beery

Parameter-efficient finetuning

- **Problem**: when the base model is large, expensive to store weights for every task.
- For example, DeepSeek V3 671B parameters $\approx$ 1.3TB in float16.
- Given pretrained weights Φ_0, learn an additive update: $\Phi_0 + \Delta\Phi(\Theta)$ parameterized by a much smaller number of parameters Θ.
- Many different solutions to this problem, such as *adapter layers* and *low rank adaptation* (LoRA).

Parameter-efficient finetuning: adapter layers

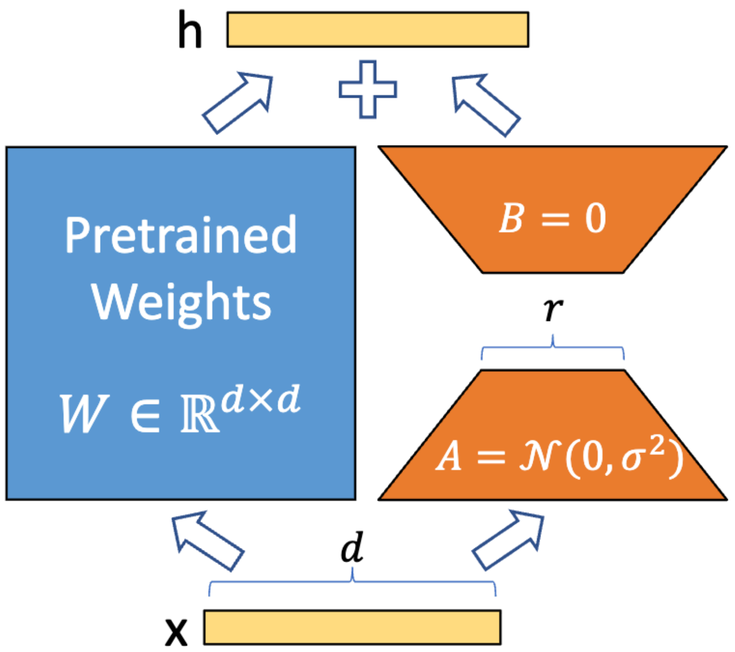




- Add small *adapter modules* to each transformer block.
- Each is represented using a relatively small number of parameters.
- Most other layers are frozen during finetuning.
- Downside: requires extra sequential computation, increasing latency.

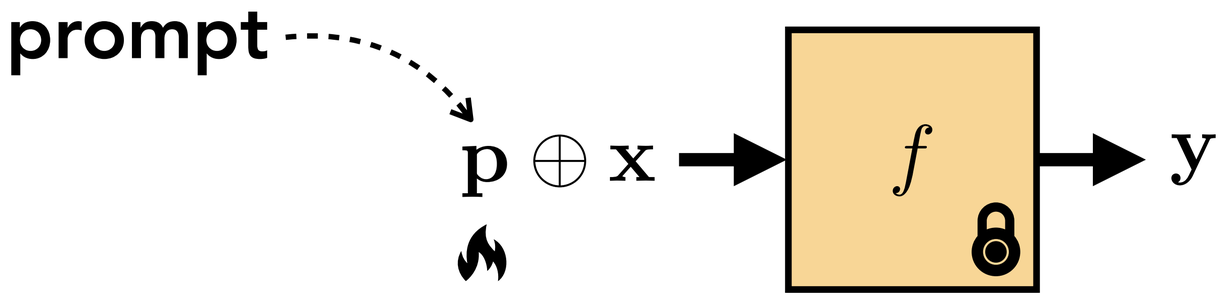
Source: [Houlsby et al., "Parameter-Efficient Transfer Learning for NLP", 2019]

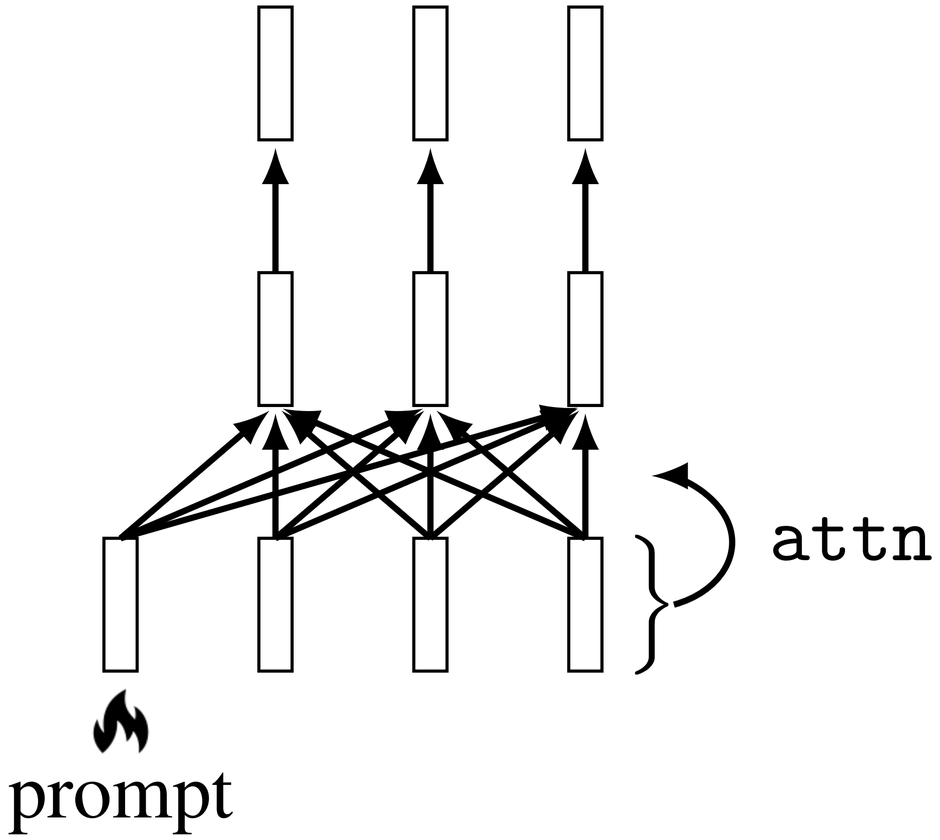
Parameter-efficient finetuning: low-rank adaptation



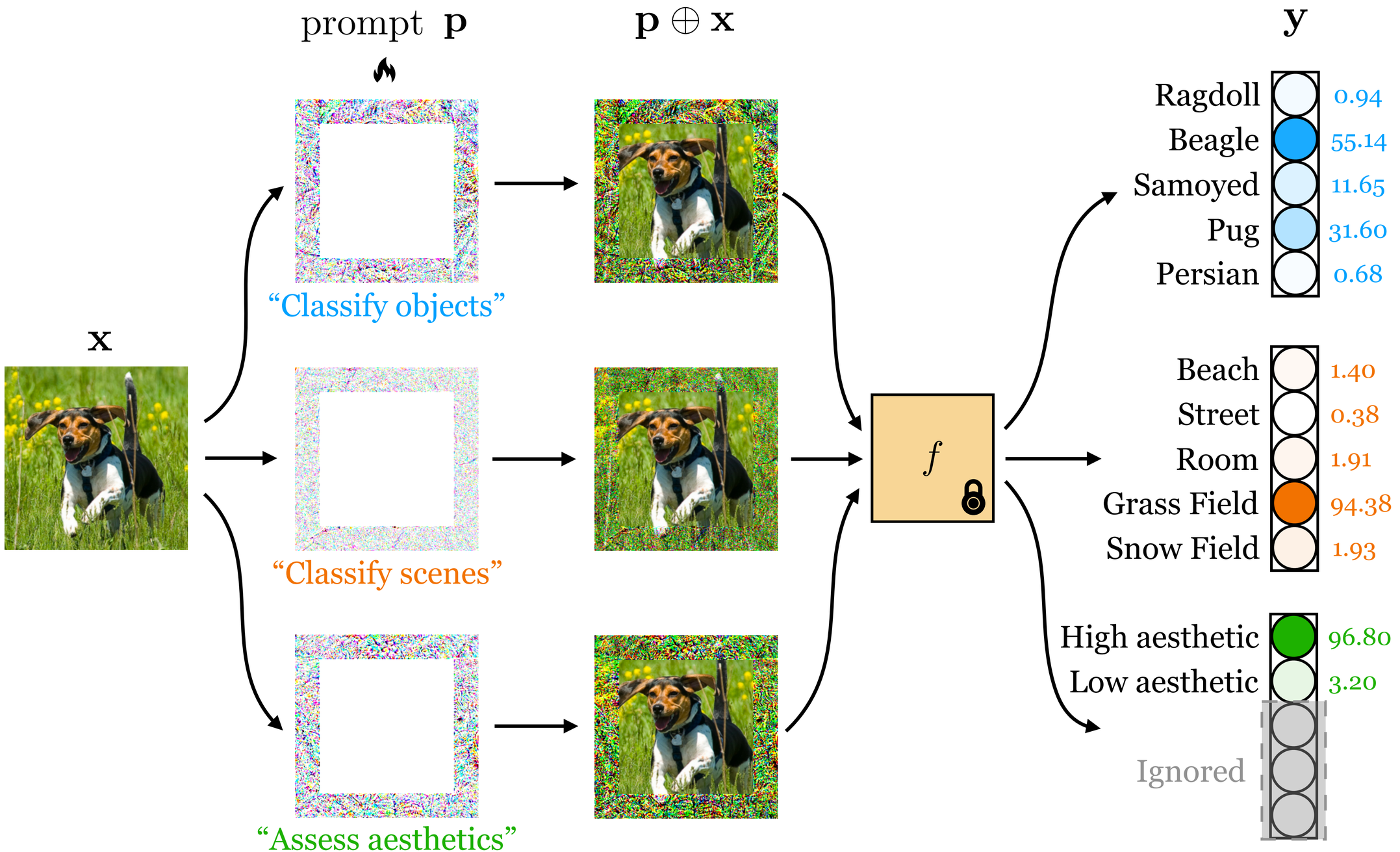
- Low-rank adaptation (LoRA) [Hu et al., 2021]: add a low-rank matrix, replacing linear layer weight matrix W_0 with:
  $W' = W_0 + BA$
  where $W_0 \in \mathbb{R}^{d \times k}$ and $B \in \mathbb{R}^{d \times r}, A \in \mathbb{R}^{r \times k}$
- To ensure low rank, $r \ll \min(d, k)$.
- Freeze W_0 during finetuning.
- At deployment time, can explicitly compute W'. Doesn't add extra computation or latency.
- In many applications, gets performance similar to full finetuning when properly tuned [Schulman 2025].

Can also treat the prompt as a learnable parameter



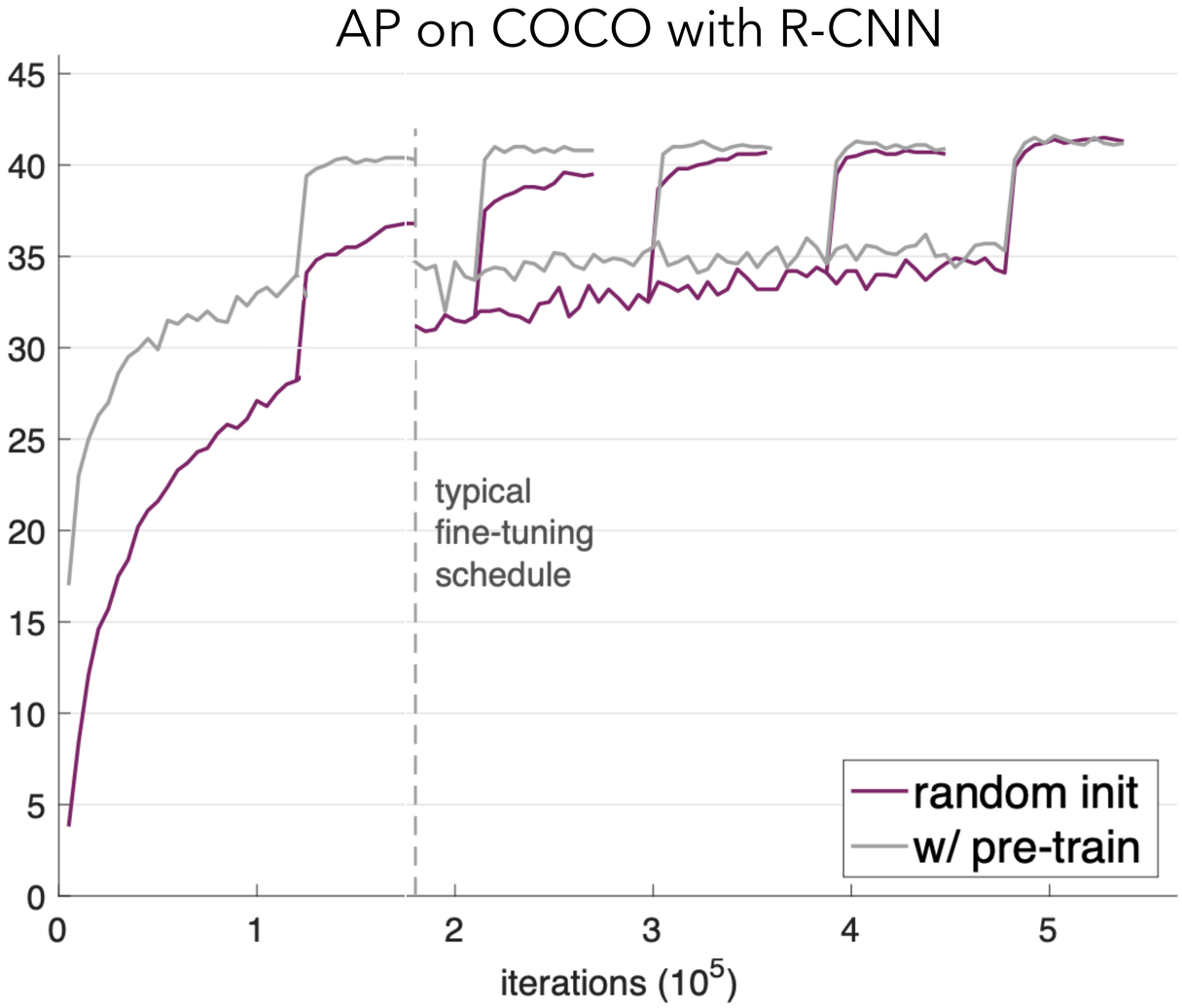


Source: Isola, Freeman, Torralba. FCV book.



Source: Isola, Freeman, Torralba. FCV book.

But training from scratch can perform well too, given enough data.



Case study: object detection

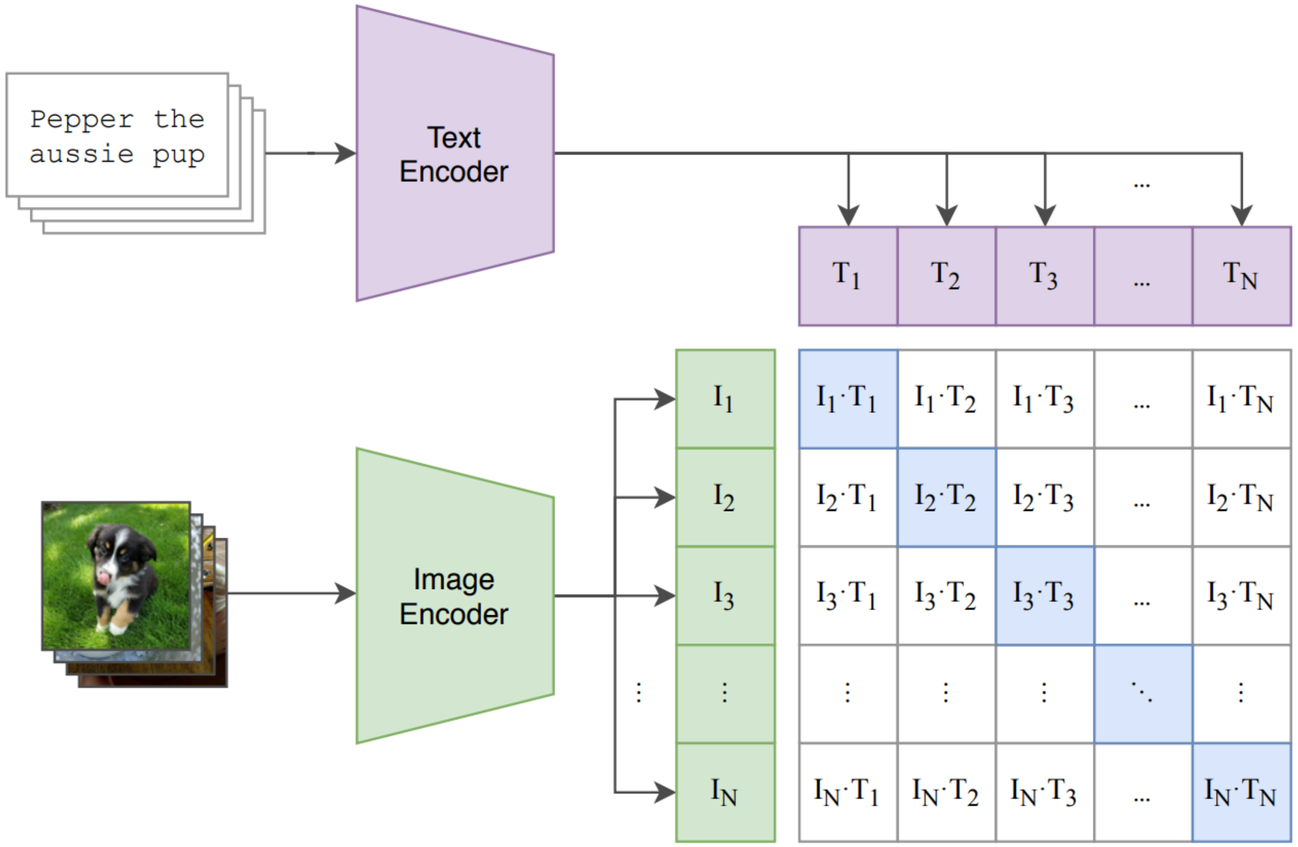
- ImageNet pretraining (a strong computer vision pretraining) speeds up object detection training by 5x
- Big performance gains for small/medium datasets (e.g. 1K examples per class)
- But no benefit for large datasets

[He et al. 2018]

What tasks do we train on?

Non-generative pretraining

Example: contrastive language-image pretraining



- Learn to associate language with images.
- Pull feature embeddings from same (image, text) pairs together. Push apart mismatched examples.
- Can't generate either signal, though learns something akin to the conditional distributions: $p(I \mid T)$ and $p(T \mid I)$.

Why *generative* pretraining?

Class discussion.

Why *generative* pretraining?

- Lots of raw text and sensory data available.
- Generating data well requires capturing interesting patterns in the data.
- Recognizing these patterns is often similar to what you do in downstream tasks.
- Many tasks require generating data.
- Unlike many other unsupervised learning tasks, no simple shortcuts.

Generative pretraining: an idea with a long history

To Recognize Shapes, First Learn to Generate Images

Geoffrey Hinton
Department of Computer Science
University of Toronto
&
Canadian Institute for Advanced Research

October 26, 2006

Semi-supervised Sequence Learning

Andrew M. Dai
Google Inc.
adai@google.com

Quoc V. Le
Google Inc.
qvl@google.com

Abstract

We present two approaches to use unlabeled data to improve Sequence Learning with recurrent networks. The first approach is to predict what comes next in a sequence, which is a language model in NLP. The second approach is to use a sequence autoencoder, which reads the input sequence into a vector and predicts the input sequence again. These two algorithms can be used as a "pretraining" algorithm for a later supervised sequence learning algorithm. In other words, the parameters obtained from the pretraining step can then be used as a starting point for other supervised training models. In our experiments, we find that long short term memory recurrent networks after pretrained with the two approaches become more stable to train and generalize better. With pretraining, we were able to achieve strong performance in many classification tasks, such as text classification with IMDB, DBpedia or image recognition in CIFAR-10.

Roles of Pre-Training and Fine-Tuning in Context-Dependent DBN-HMMs for Real-World Speech Recognition

Dong Yu, Li Deng
Microsoft Research
One Microsoft Way
Redmond, WA 98052, USA
{dongyu,deng}@microsoft.com

George E. Dahl
Department of Computer Science
University of Toronto
Ontario, Canada
gdahl@cs.toronto.edu

Abstract

Recently, deep learning techniques have been successfully applied to automatic speech recognition tasks -- first to phonetic recognition with context-independent deep belief network (DBN) hidden Markov models (HMMs) and later to large vocabulary continuous speech recognition using context-dependent (CD) DBN-HMMs. In this paper, we report our most recent experiments designed to understand the roles of the two main phases of the DBN learning -- pre-training and fine tuning -- in the recognition performance of a CD-DBN-HMM based large-vocabulary speech recognizer. As expected, we show that pre-training can initialize weights to a point in the space where fine-tuning can be effective and thus is crucial in training deep structured models. However, a moderate increase of the amount of unlabeled pre-training data has an insignificant effect on the final recognition results as long as the original training size is sufficiently large to initialize the DBN weights. On the other hand, with additional labeled training data, the fine-tuning phase of DBN training can significantly improve the recognition accuracy.

Generative pretraining

Is generative modeling a good pretraining task?

Improving Language Understanding by Generative Pre-Training

Alec Radford
OpenAI
alec@openai.com

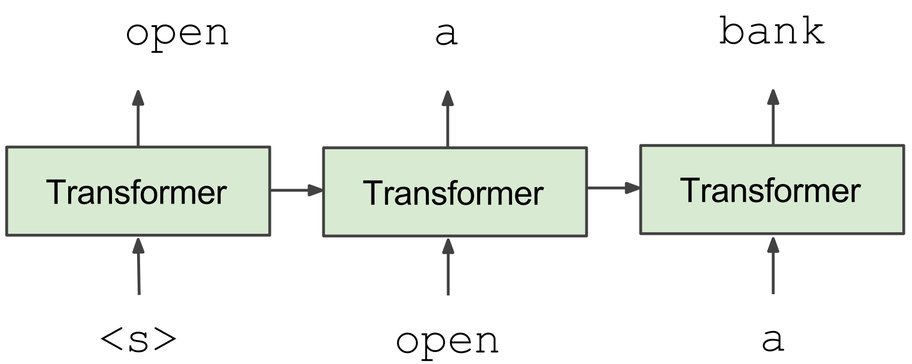
Karthik Narasimhan
OpenAI
karthikn@openai.com

Tim Salimans
OpenAI
tim@openai.com

Ilya Sutskever
OpenAI
ilyasu@openai.com

Finetuning GPT

GPT training



Finetuning for classification task

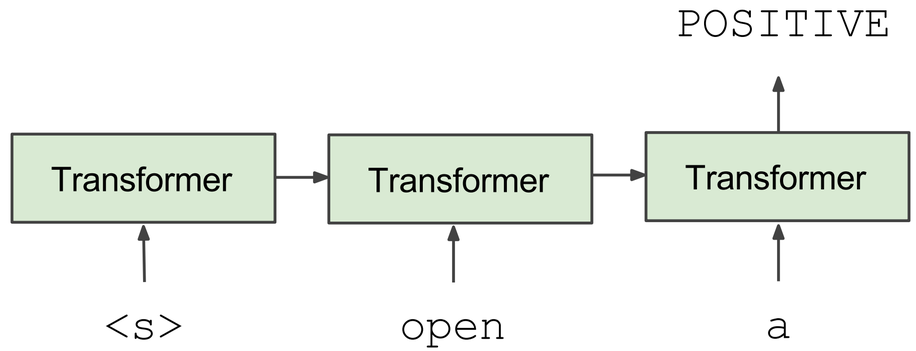
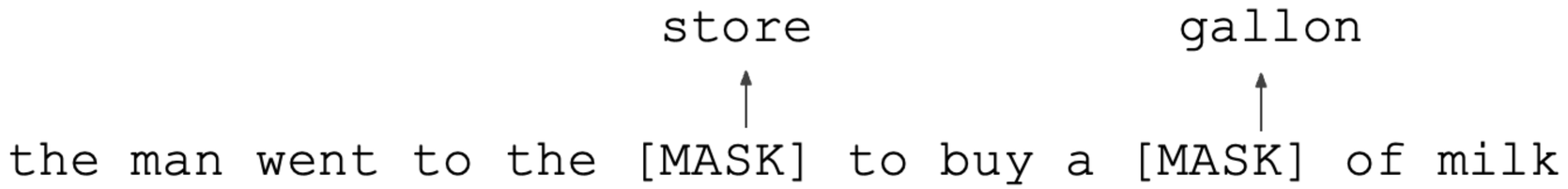


Figure source: Jacob Devlin BERT slides

Limitations

- GPT is a causal model: we predict text left-to-right,
- Attention is unidirectional
- Only learns the conditional distributions from autoregressive training, i.e., $p(x_t \mid x_1, x_2, \ldots, x_{t-1})$.

Masked language modeling (MLM)



- "Fill in the gaps". Remove tokens at random, introducing special [MASK] tokens.
  - Sometimes introduce random words instead of [MASK]
- BERT (Bidirectional Encoder Representations from Transformers) masks 15% of tokens.
- Learns conditional distribution: $\prod_{i \in M} p(x_i \mid \{x_j \mid j \notin M\})$.
- Doesn't aim to directly generate text (though still possible if you're careful, as we discussed in guest lecture on masked diffusion language models!).

Figure source: Jacob Devlin BERT slides

MLM vs. causal model finetuning

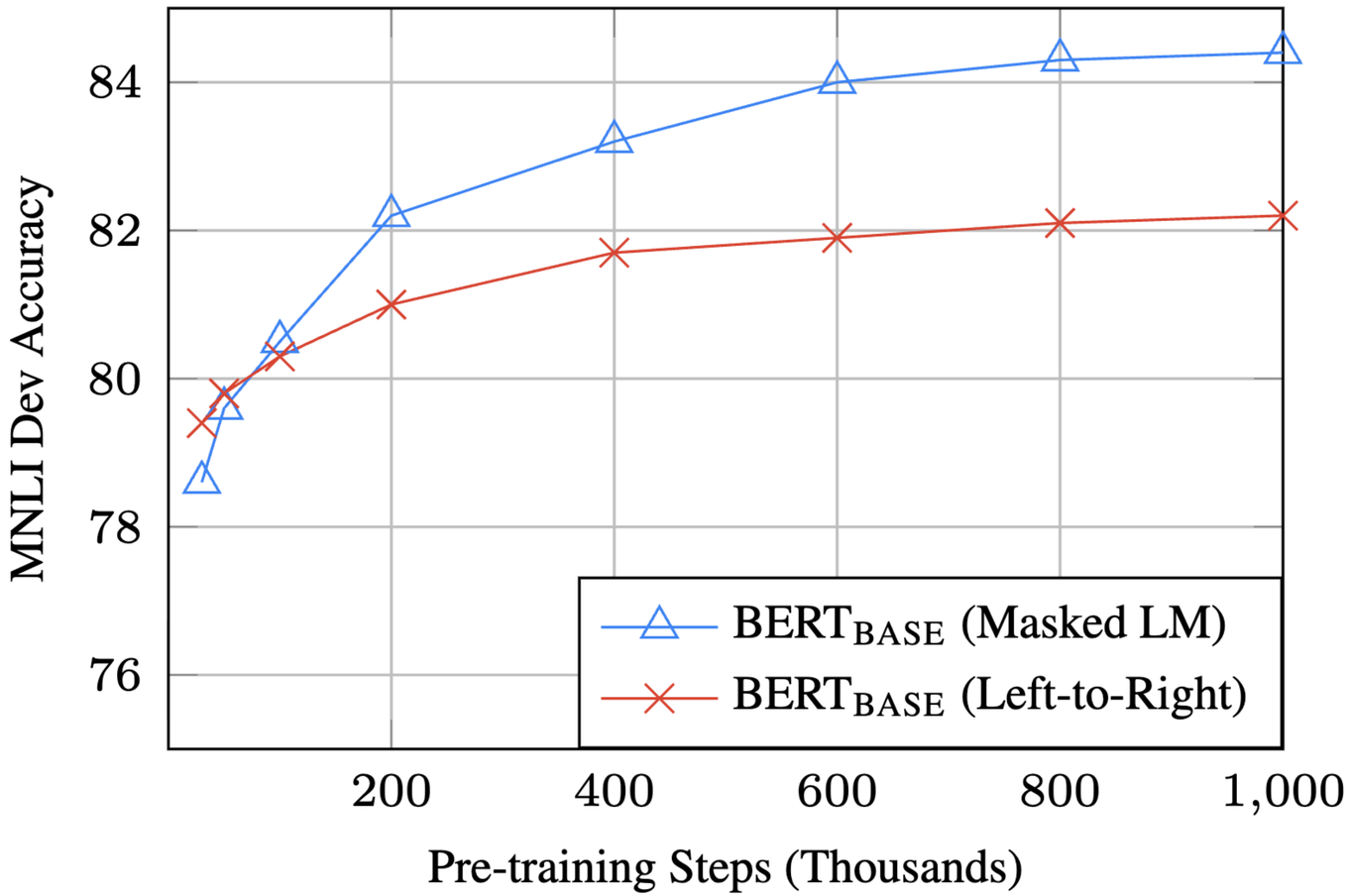
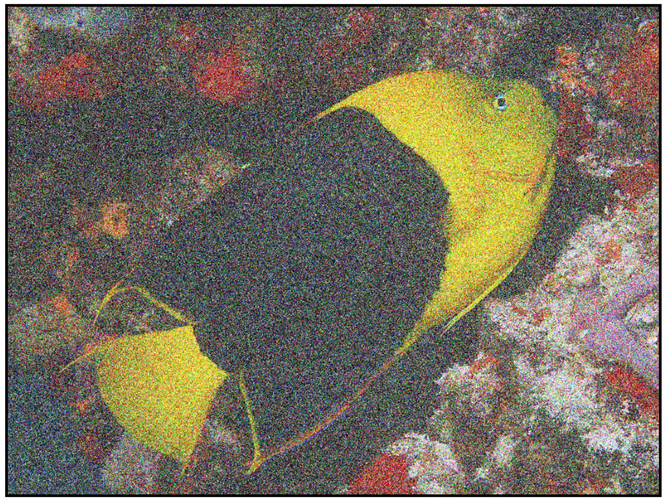
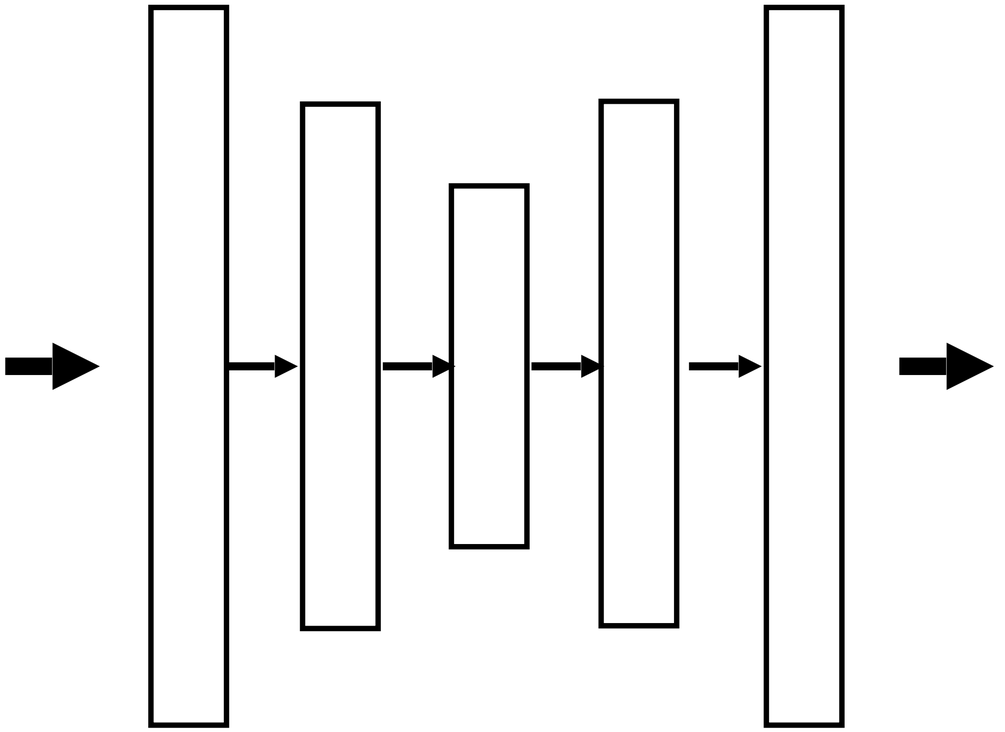


Figure source: Jacob Devlin and BERT paper

In vision: denoising autoencoder



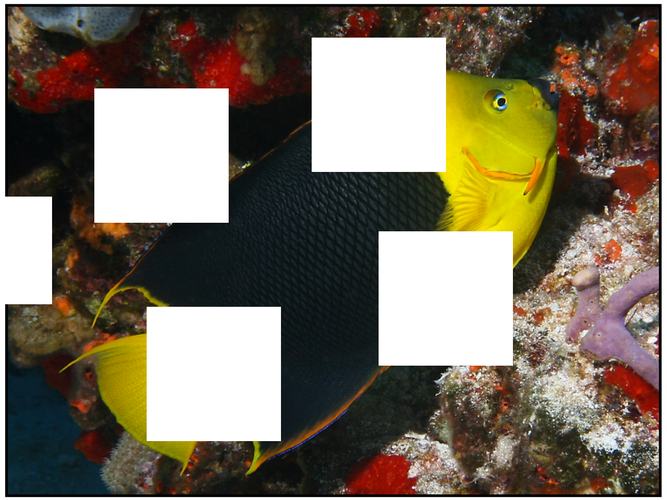
Noisy image



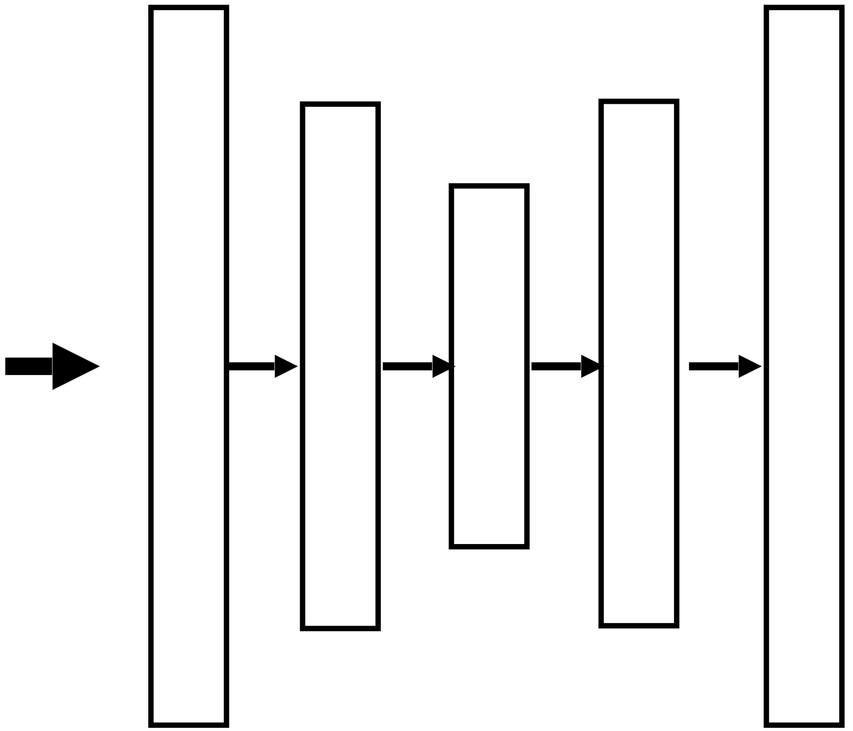
Reconstructed image

[Vincent et al., 2008]

In vision: denoising autoencoder



Noisy example

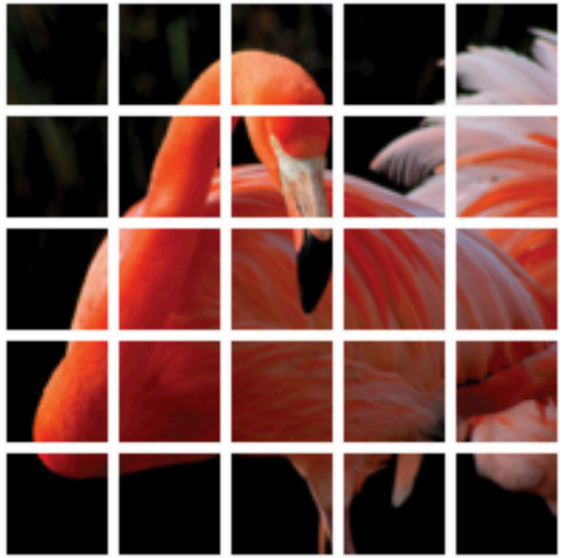


Reconstructed example

Other types of “noise”.

[Vincent et al., 2008, Pathak et al. 2015, He, 2020]

Masked autoencoders



image

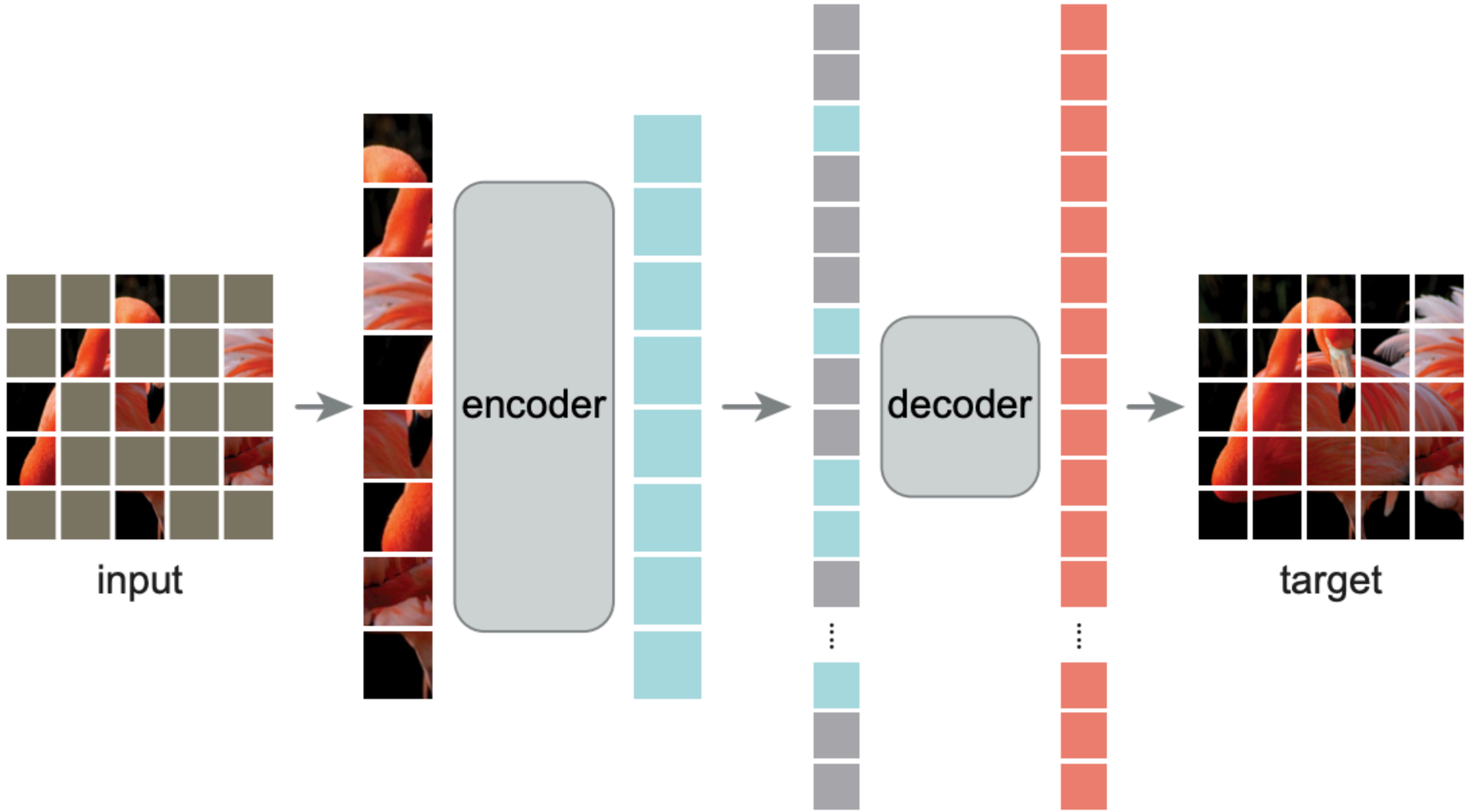
[Kaiming He et al., “Masked Autoencoders Are Scalable Vision Learners”, 2021]

Masked autoencoders

masked image

[Kaiming He et al., “Masked Autoencoders Are Scalable Vision Learners”, 2021]

Masked autoencoders with transformers



Autoencoder loss:

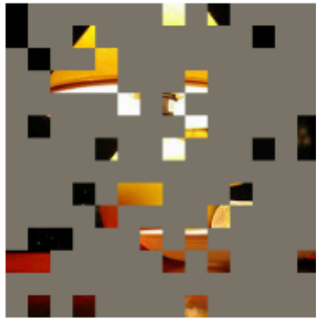
$$\mathscr{L}_\theta = \|\mathbf{X} - \hat{\mathbf{X}}\|^2$$

[Kaiming He et al., "Masked Autoencoders Are Scalable Vision Learners", 2021]

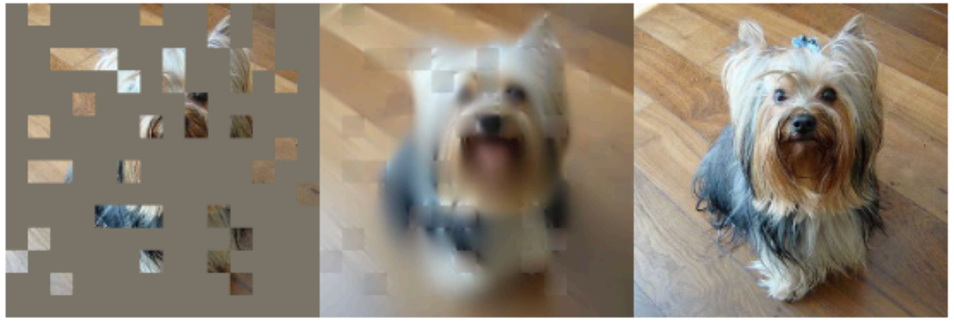
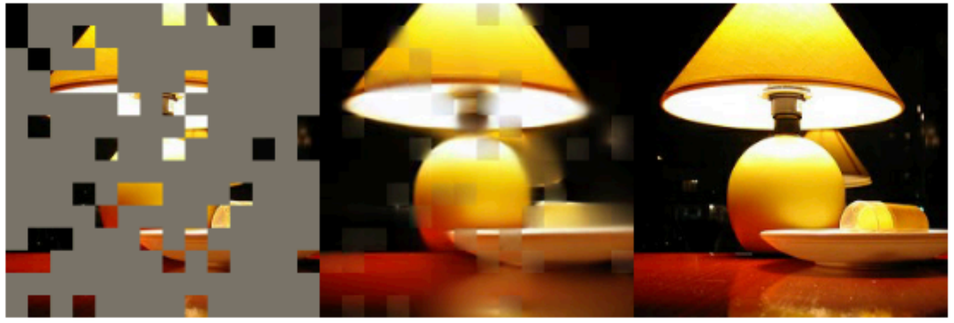
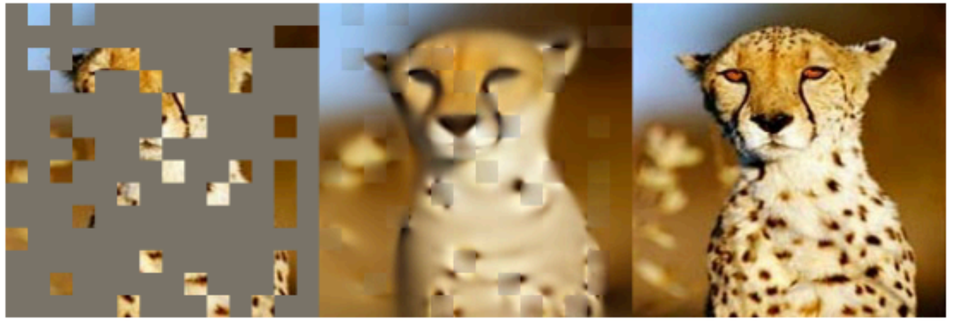


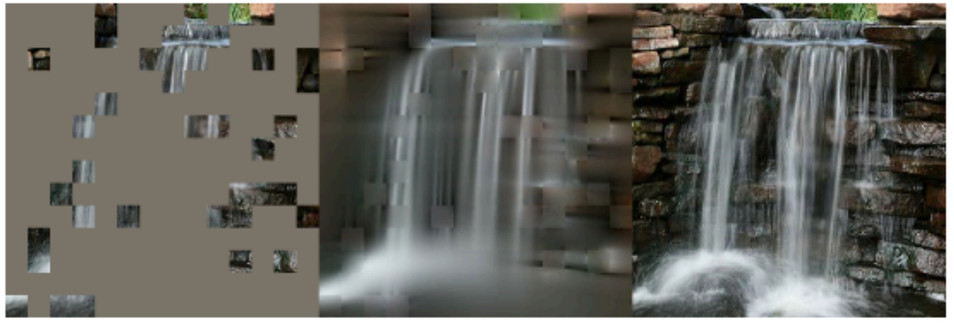
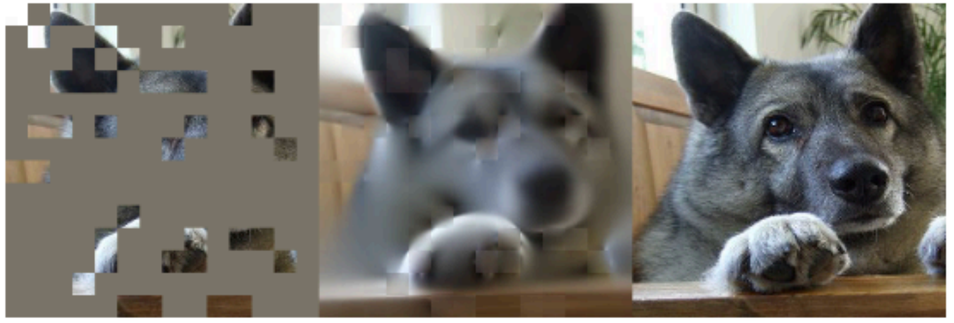




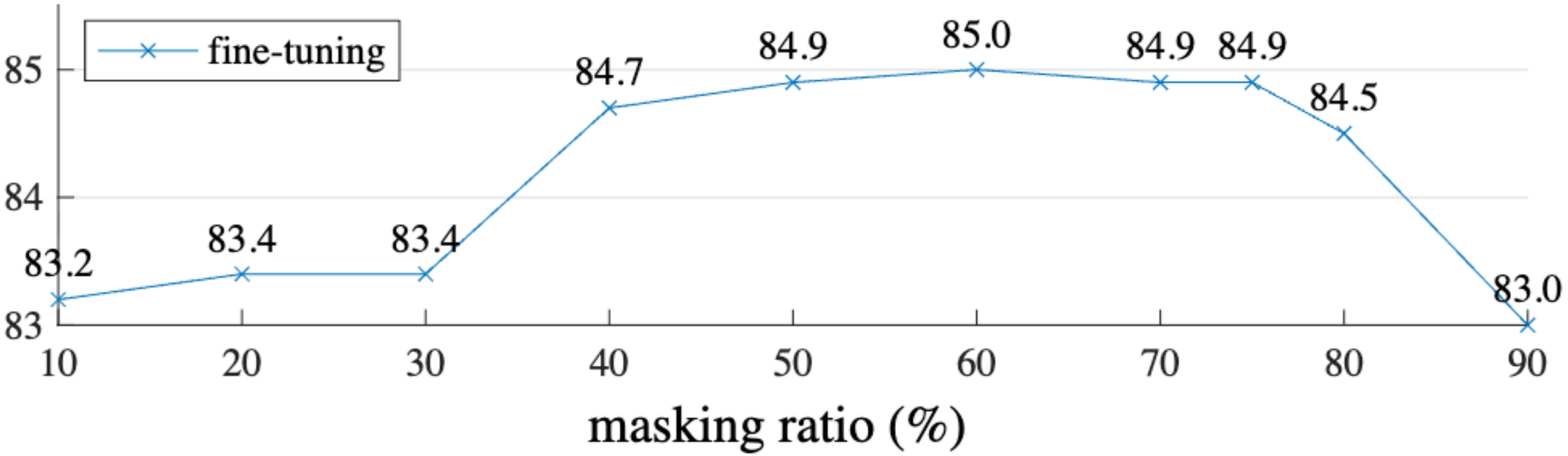




Feature learning performance



Downstream ImageNet recognition performance

Zero-shot learning

”I’m not the cleverest man in the world, but like they say in French: **Je ne suis pas un imbecile [I’m not a fool].**

In a now-deleted post from Aug. 16, Soheil Eid, Tory candidate in the riding of Joliette, wrote in French: ”**Mentez mentez, il en restera toujours quelque chose**,” which translates as, ”**Lie lie and something will always remain.**”

“I hate the word ‘**perfume**,”’ Burr says. ‘It’s somewhat better in French: ‘**parfum**.’

If listened carefully at 29:55, a conversation can be heard between two guys in French: “**-Comment on fait pour aller de l’autre coté? -Quel autre coté?**”, which means “**- How do you get to the other side? - What side?**”.

If this sounds like a bit of a stretch, consider this question in French: **As-tu aller au cinéma?**, or **Did you go to the movies?**, which literally translates as Have-you to go to movies/theater?

“**Brevet Sans Garantie Du Gouvernement**”, translated to English: “**Patented without government warranty**”.

Naturally occurring translation demonstrations in internet text

Source: [Radford et al., GPT-2, 2019]

Prompting

- Edit the input to a model to change its behavior.
- Common in models that take textual instructions as input.
- Can also be used for models that take other inputs, such as image inputs.

Source: Isola, Freeman, Torralba. FCV book.

Prompting

Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

```
1  Translate English to French:        ← task description
2  cheese =>                           ← prompt
```

Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

```
1  Translate English to French:        ← task description
2  sea otter => loutre de mer          ← examples
3  peppermint => menthe poivrée        ←
4  plush girafe => girafe peluche      ←
5  cheese =>                           ← prompt
```

In-context learning: provide training examples in a language model's context window (rather than via finetuning).

Source: [Brown et al., GPT-3]

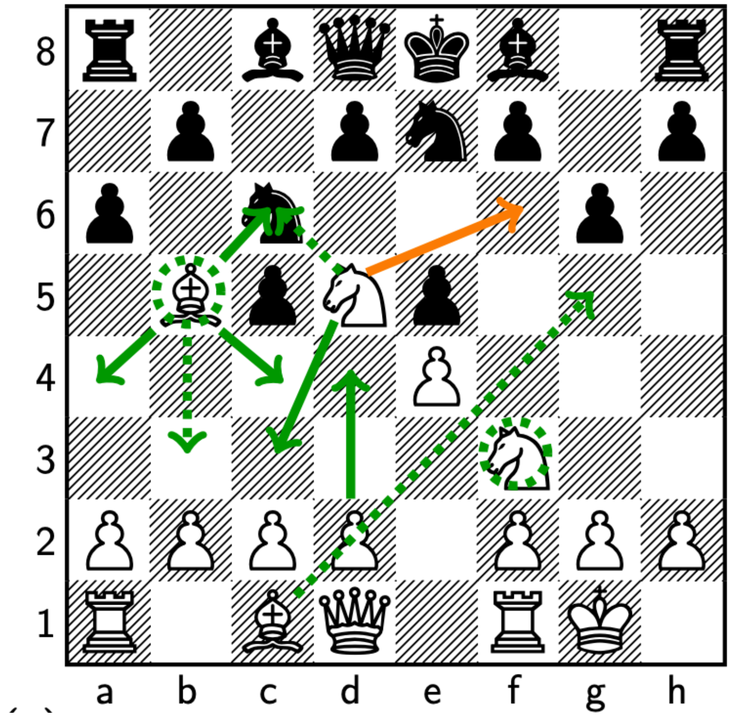
Zero-shot prediction improves with model size

Q: What movie does this emoji describe? 👧🐟🐠🐡

2m: i'm a fan of the same name, but i'm not sure if it's a good idea
16m: the movie is a movie about a man who is a man who is a man …
53m: the emoji movie 🐟🐠🐡
125m: it's a movie about a girl who is a little girl
244m: the emoji movie
422m: the emoji movie
1b: the emoji movie
2b: the emoji movie
4b: the emoji for a baby with a fish in its mouth
8b: the emoji movie
27b: the emoji is a fish
128b: finding nemo

Source: [Srivastava et al., BIG-bench 2022]

Prompting

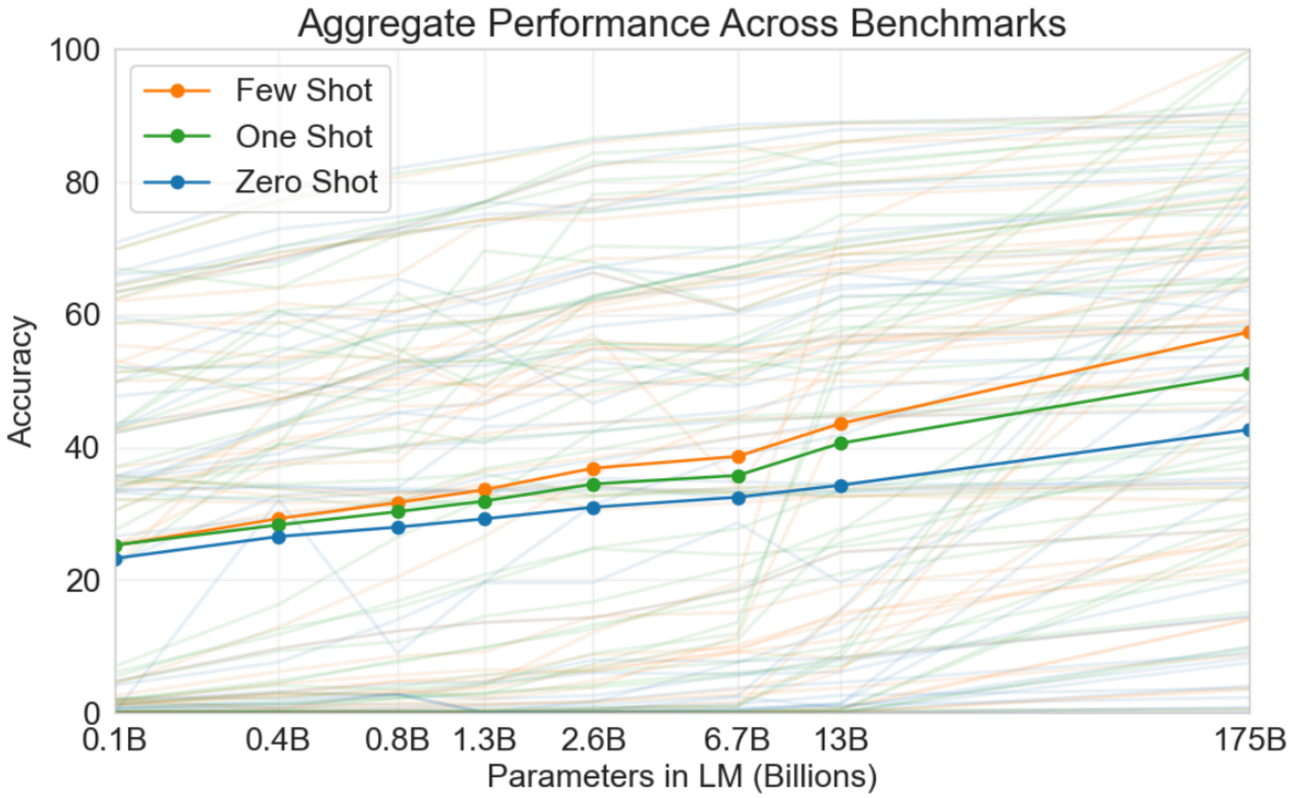


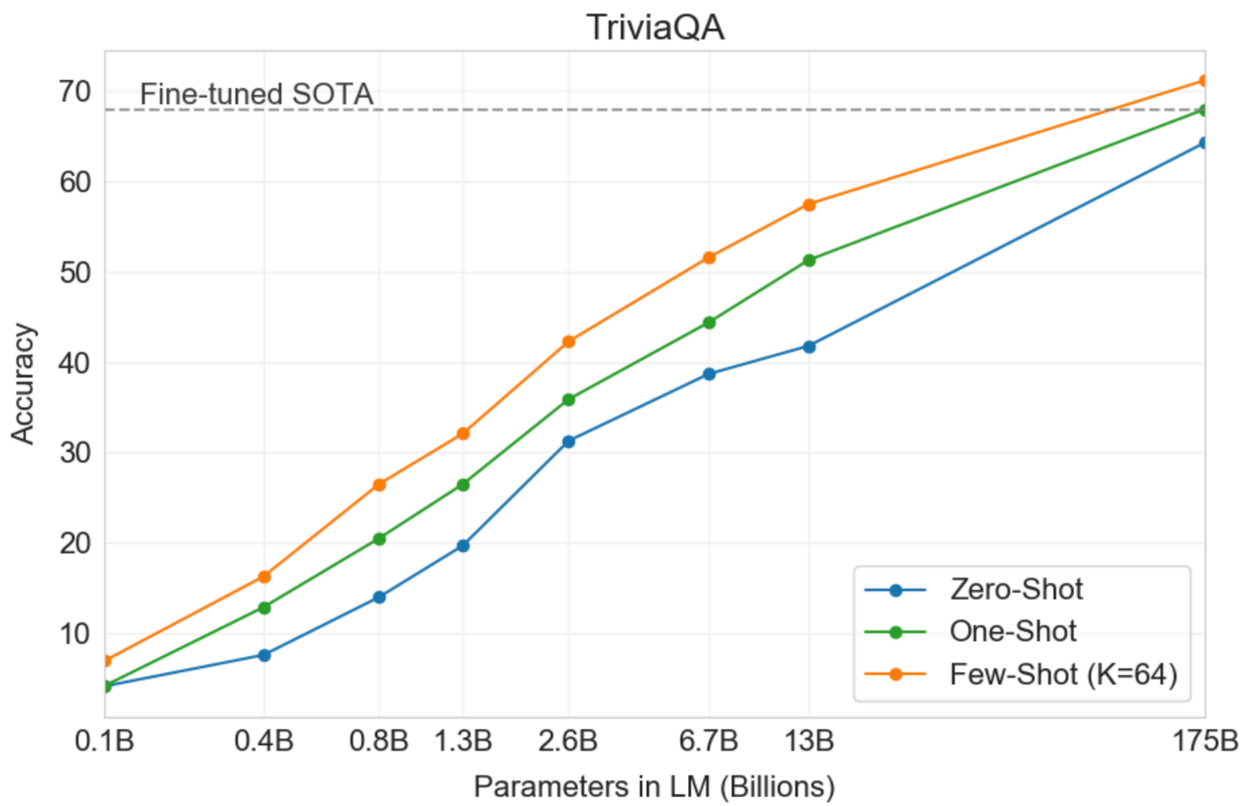

```
In the following chess position, find a checkmate-in-one move.
1.  e4 c5 2.  Nf3 e5 3.  Nc3 Nc6 4.  Bb5 Nge7 5.  O-O g6 6.  Nd5 a6 7.

Nf6#,
```


Source: [Srivastava et al., BIG-bench 2022]

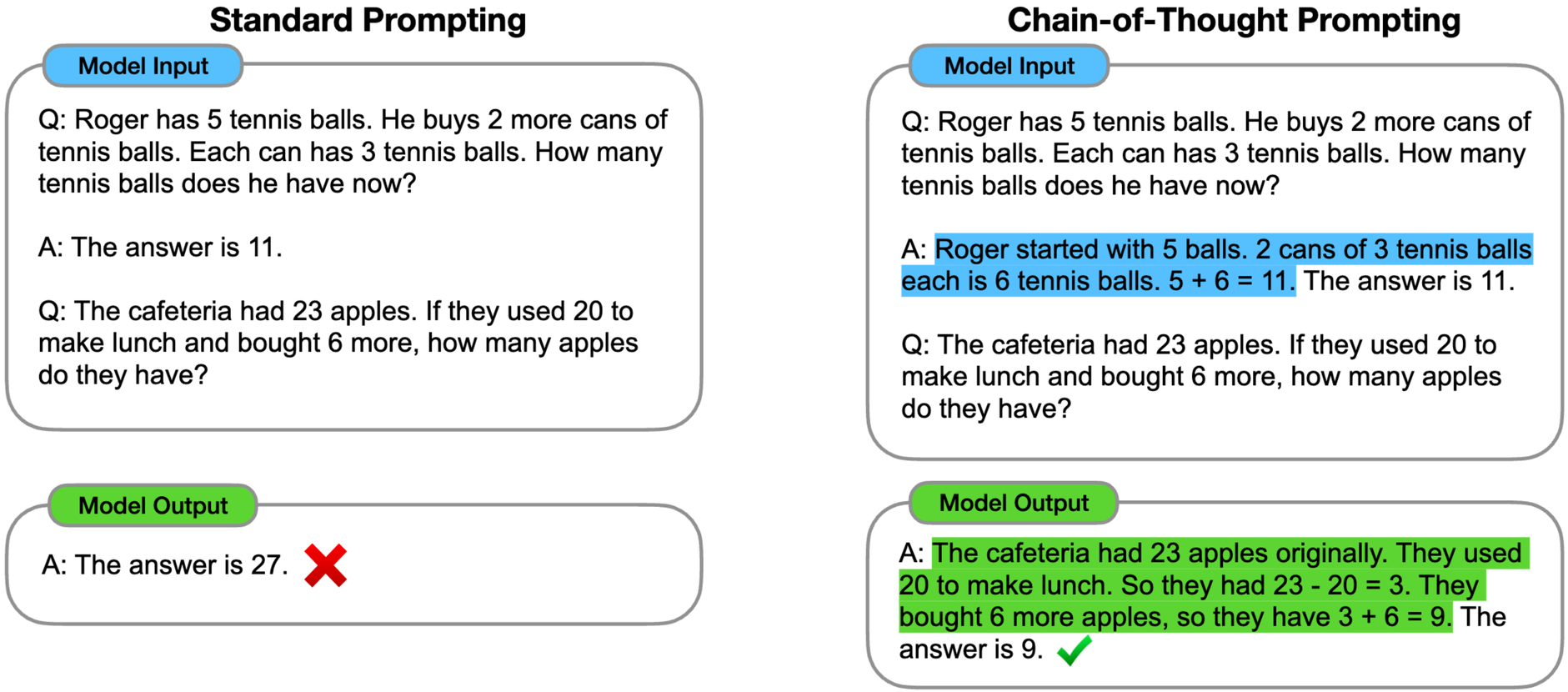
Prompting





Source: [Brown et al., GPT-3]

How much does the prompt matter?



Give examples of reasoning in the prompt, making it more likely for "complete-the-sequence" to yield the right answer.

Source: [Wei et al., "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models", 2022]

How much does the prompt matter?

Zero-shot

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?
A: The answer (arabic numerals) is

(Output) 8 ✗

Zero-shot-CoT (Ours)

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?
A: ***Let's think step by step.***

(Output) There are 16 balls in total. Half of the balls are golf balls. That means that there are 8 golf balls. Half of the golf balls are blue. That means that there are 4 blue golf balls. ✓

Simply adding the text "Let's think step by step" has a similar effect!

Source: [Kojima et al., "Large Language Models are Zero-Shot Reasoners", 2022]

What else can video models do?

Object segmentation

"Create an animation of instance segmentation being performed on this photograph: each distinct entity is overlaid in a different flat color.
Scene:
• The animation starts from the provided, unaltered photograph.
• The scene in the photograph is static and doesn't move.
• First, the background fades to {white, green}.
• Then, the first entity is covered by a flat color, perfectly preserving its silhouette.
• Then the second entity, too, is covered by a different flat color, perfectly preserving its silhouette.
• One by one, each entity is covered by a different flat color.
• Finally, all entities are covered with different colors.
Camera:
• Static shot without camera movement.
• No pan.
• No rotation.
• No zoom.
• No glitches or artifacts."

From [Wiedemer et al., "Video models are zero-shot learners and reasoners", 2025]

Object segmentation

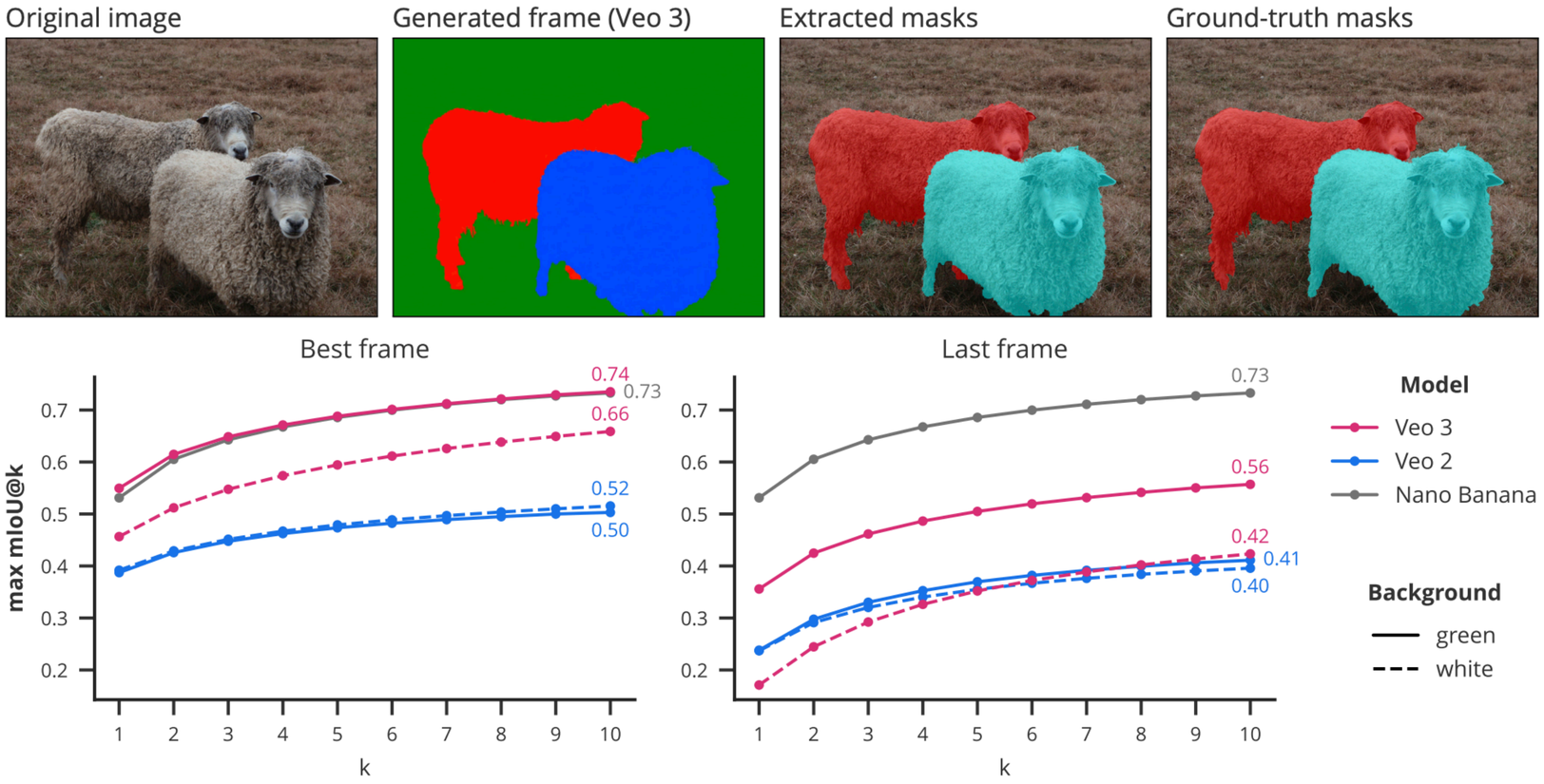
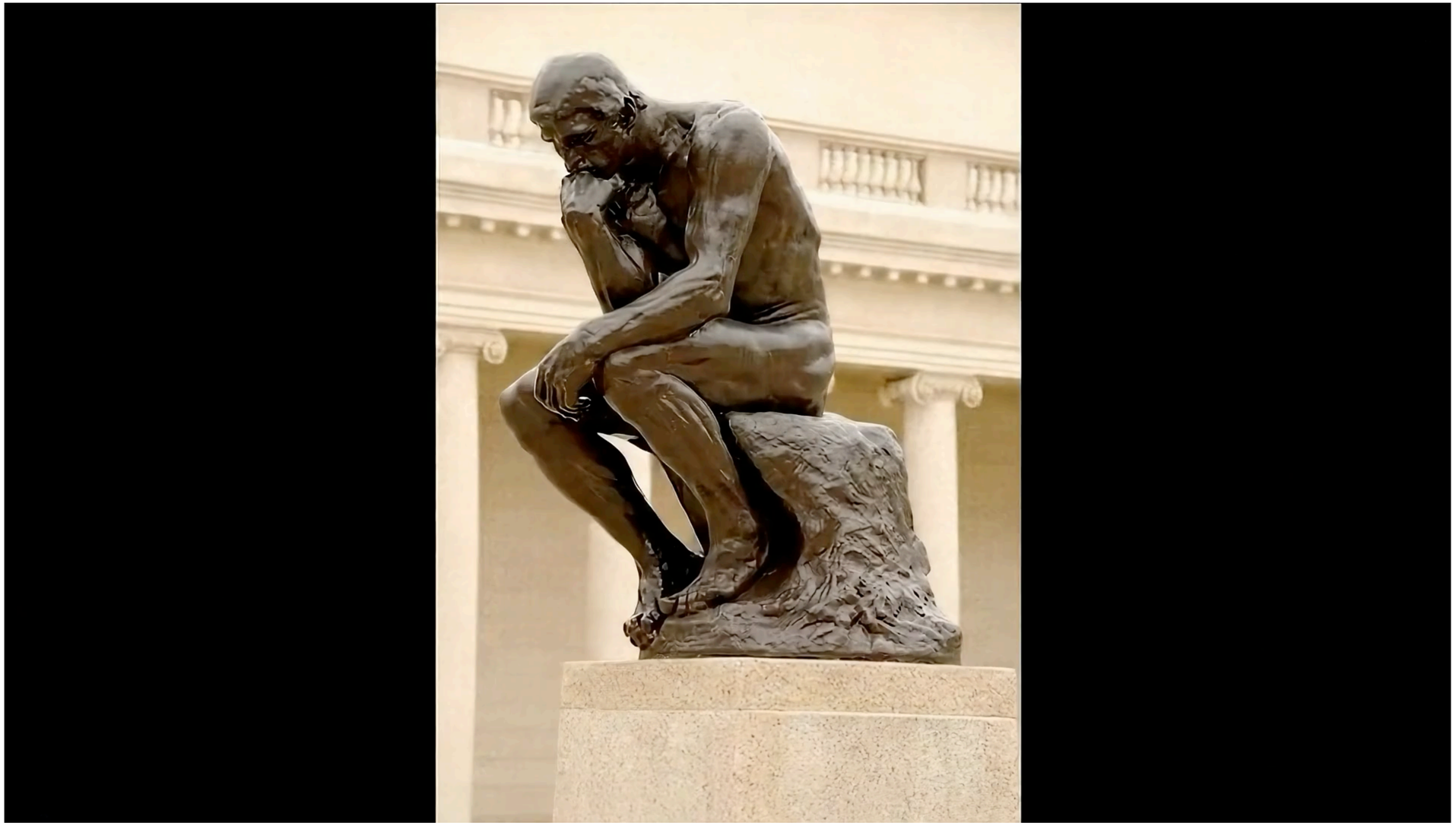


Figure 4 | **Class-agnostic instance segmentation** on a subset of 50 easy images (1-3 large objects) from LVIS [61]. Prompt: *"[...] each distinct entity is overlaid in a different flat color [...] the background fades to {white, green} [...]"* Details & full prompt: Sec. B.2.

Not extremely accurate…

From [Wiedemer et al., "Video models are zero-shot learners and reasoners", 2025]

Novel view synthesis

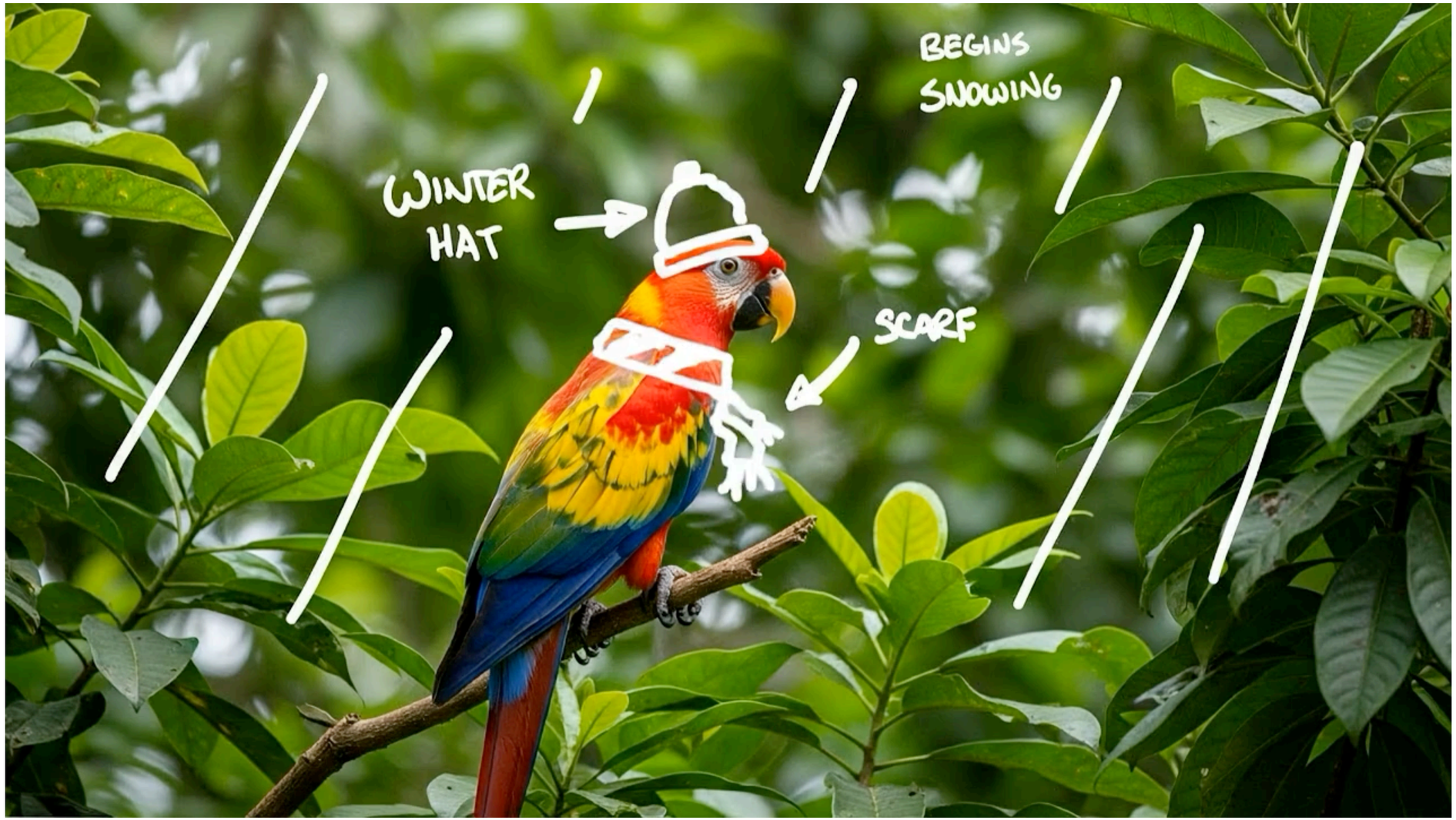


From [Wiedemer et al., “Video models are zero-shot learners and reasoners”, 2025]

Colorization

From [Wiedemer et al., “Video models are zero-shot learners and reasoners”, 2025]

“Doodle-based” editing

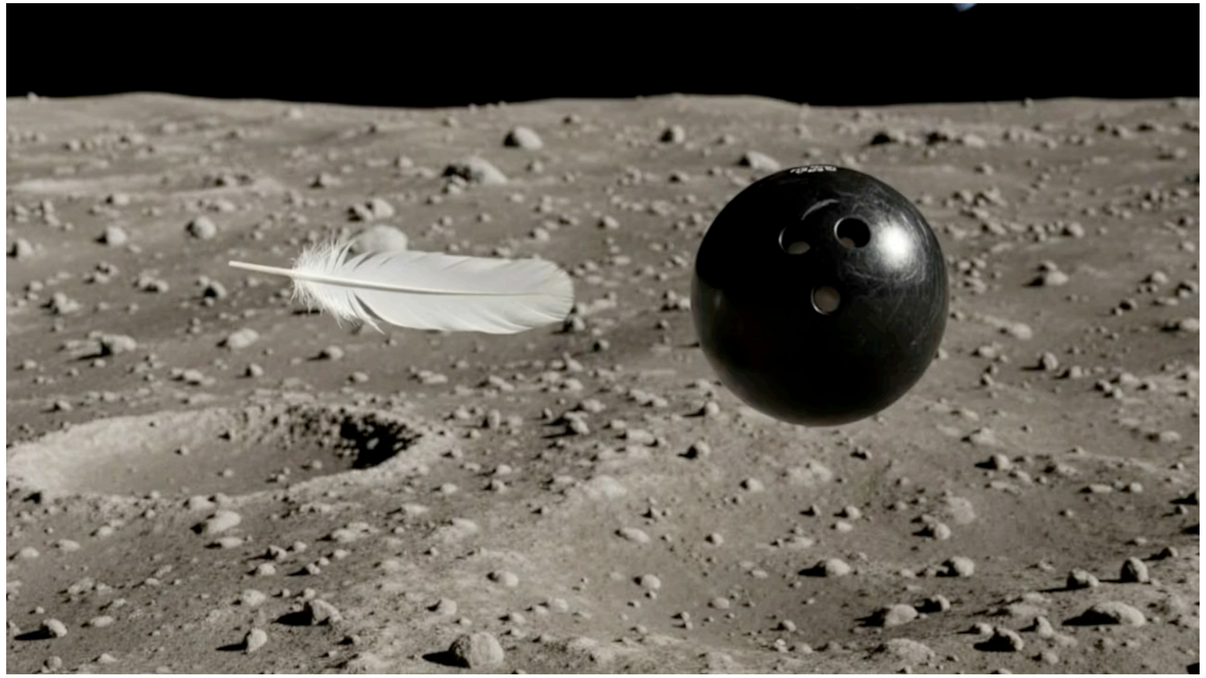


From [Wiedemer et al., “Video models are zero-shot learners and reasoners”, 2025]

Intuitive physics

From [Wiedemer et al., “Video models are zero-shot learners and reasoners”, 2025]

Intuitive physics



From [Wiedemer et al., “Video models are zero-shot learners and reasoners”, 2025]

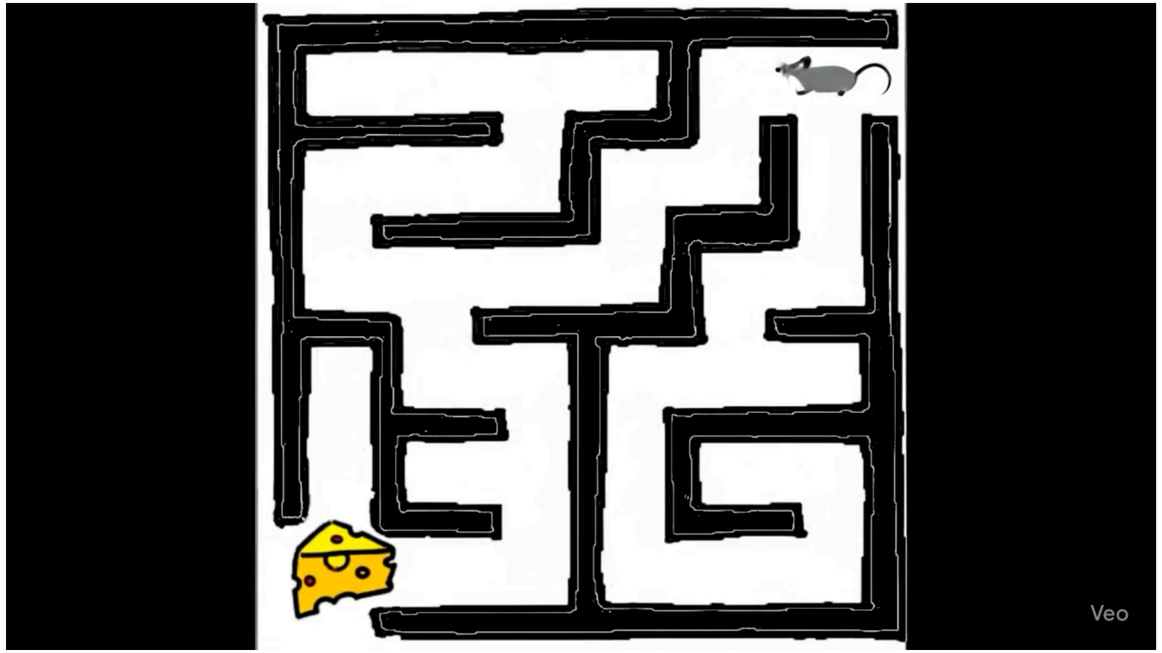
Visual Jenga

"A hand quickly removes each of the items in this image, one at a time."

From [Wiedemer et al., “Video models are zero-shot learners and reasoners”, 2025]

Task from [Bhattad, et al. “Visual Jenga: Discovering Object Dependencies via Counterfactual Inpainting”, 2025]

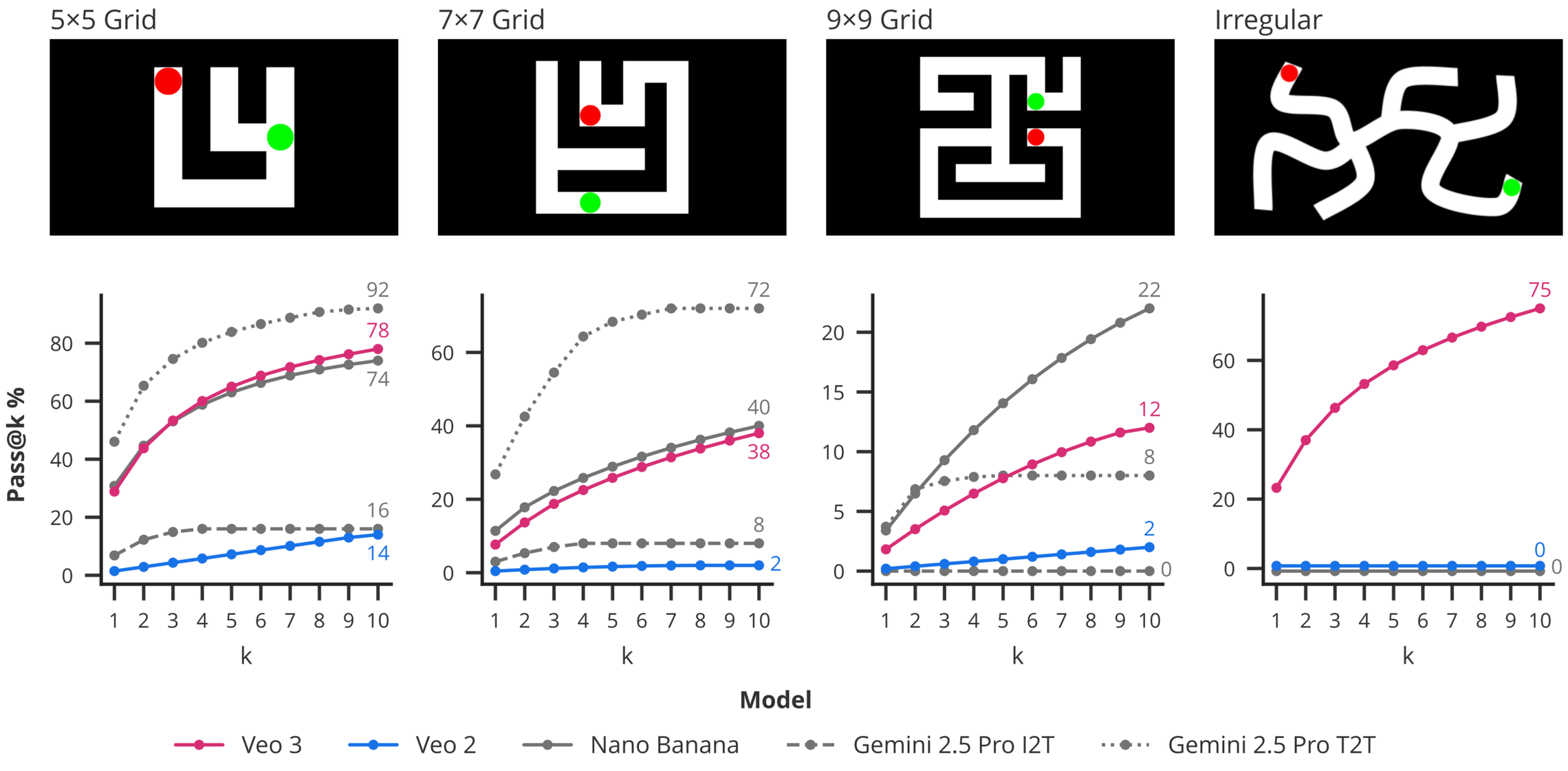
Solving mazes



"Without crossing any black boundary, the grey mouse from the corner skillfully navigates the maze by walking around until it finds the yellow cheese."

From [Wiedemer et al., "Video models are zero-shot learners and reasoners", 2025]

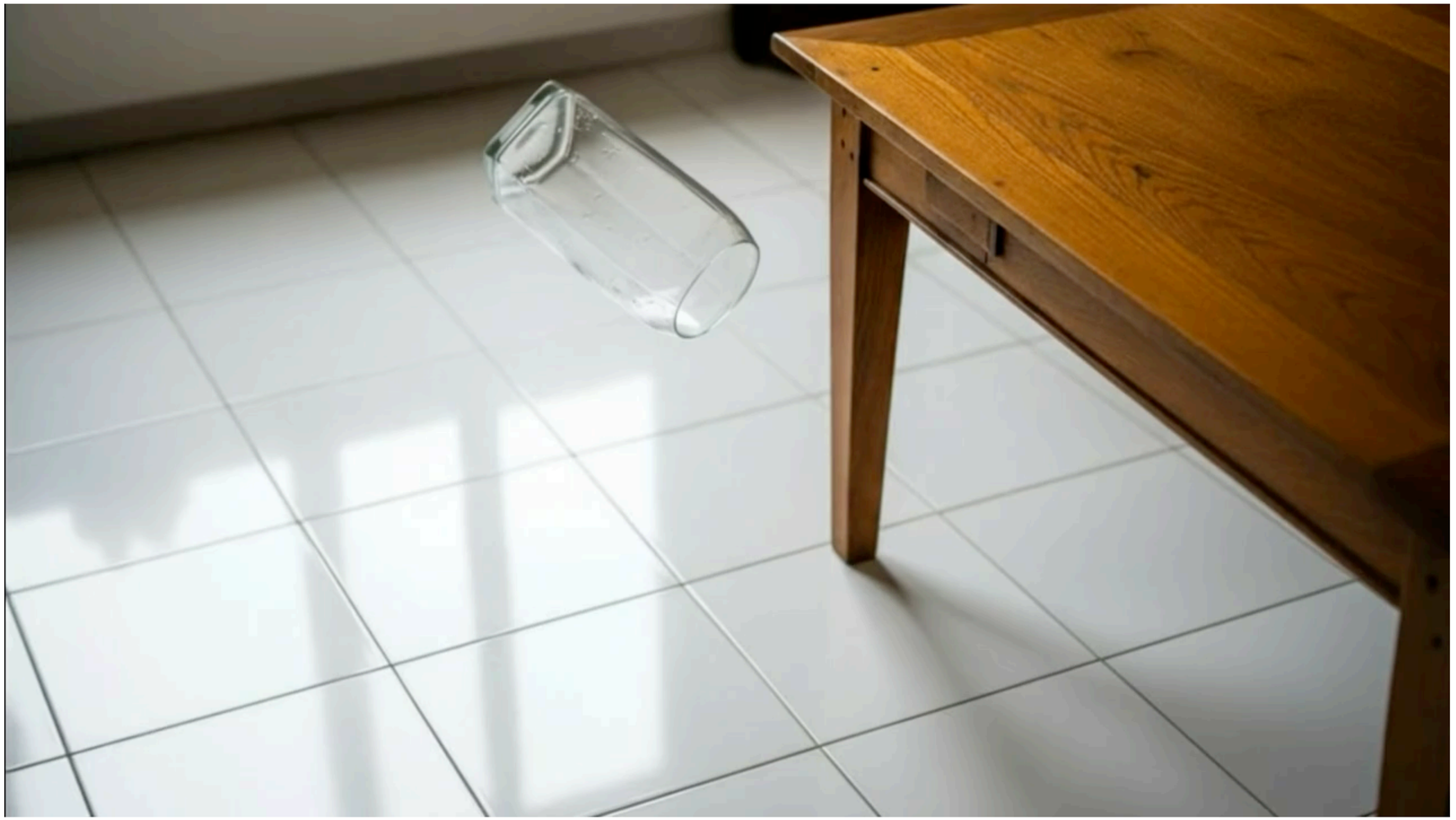
Accuracy with different models



Failure cases

From [Wiedemer et al., “Video models are zero-shot learners and reasoners”, 2025]

Failure cases



From [Wiedemer et al., “Video models are zero-shot learners and reasoners”, 2025]

Do we need prompting?

- Clever prompting can only take you so far.
- Later in the class, we'll talk about *post-training* methods that adapt generative models in other ways, beyond finetuning.

Next class: Diffusion for image manipulation