10-423/623: Generative AI Lecture 9 – In-context Learning

Henry Chai & Matt Gormley

9/25/24

Front Matter

- Announcements:
  - HW2 released ~~9/23~~ 9/24, due 10/7 at 11:59 PM
  - Quiz 2 moved to 9/30 (Monday)

Recap of…

DIFFUSION MODELS

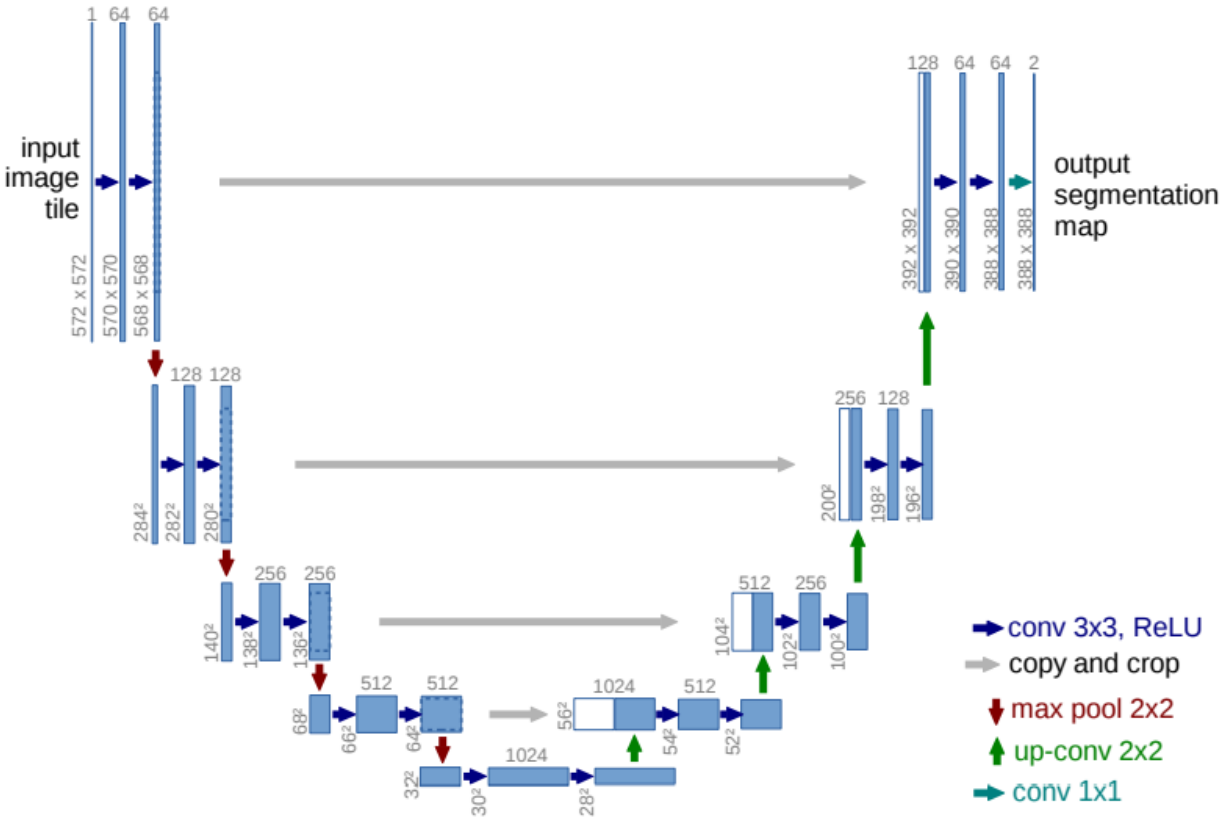
U-Net

Contracting path

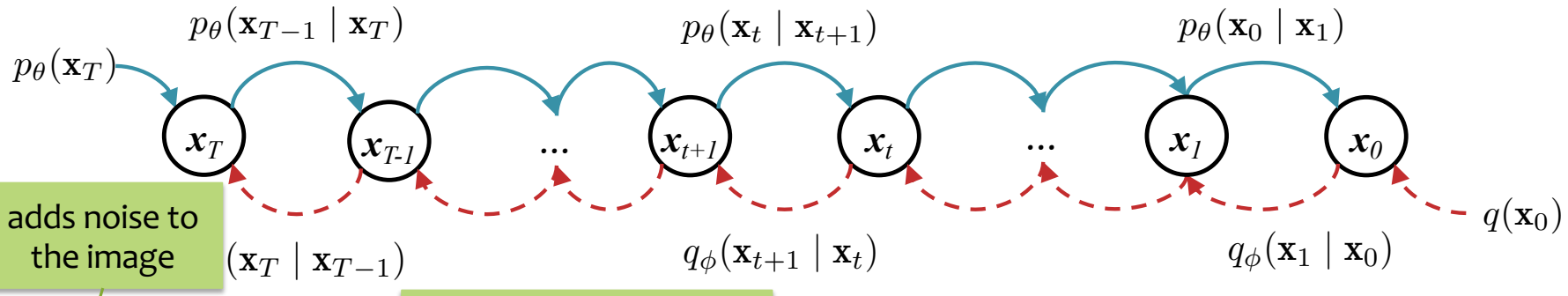
- block consists of:
  - 3x3 convolution
  - 3x3 convolution
  - ReLU
  - max-pooling with stride of 2 (downsample)
- repeat the block N times, doubling number of channels

Expanding path

- block consists of:
  - 2x2 convolution (upsampling)
  - concatenation with contracting path features
  - 3x3 convolution
  - 3x3 convolution
  - ReLU
- repeat the block N times, halving the number of channels



Diffusion Model



Forward Process:

if we could sample from this we'd be done

$$q_\phi(\mathbf{x}_{0:T}) = q(\mathbf{x}_0) \prod_{t=1}^{T} q_\phi(\mathbf{x}_t \mid \mathbf{x}_{t-1})$$

(Learned) Reverse Process:

removes noise

$$p_\theta(\mathbf{x}_{0:T}) = p_\theta(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$$

goal is to learn this

(Exact) Reverse Process:

$$q_\phi(\mathbf{x}_{0:T}) = q_\phi(\mathbf{x}_T) \prod_{t=1}^{T} q_\phi(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$$

The *exact* reverse process requires inference. And, even though $q_\phi(\mathbf{x}_t \mid \mathbf{x}_{t-1})$ is simple, computing $q_\phi(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$ is intractable! Why? Because $q(\mathbf{x}_0)$ might be not-so-simple.

$$q_\phi(\mathbf{x}_{t-1} \mid \mathbf{x}_t) = \frac{\int_{\mathbf{x}_{0:t-2,t+1:T}} q_\phi(\mathbf{x}_{0:T}) d\mathbf{x}_{0:t-2,t+1:T}}{\int_{\mathbf{x}_{0:t-2,t:T}} q_\phi(\mathbf{x}_{0:T}) d\mathbf{x}_{0:t-2,t:T}}$$

Diffusion Model

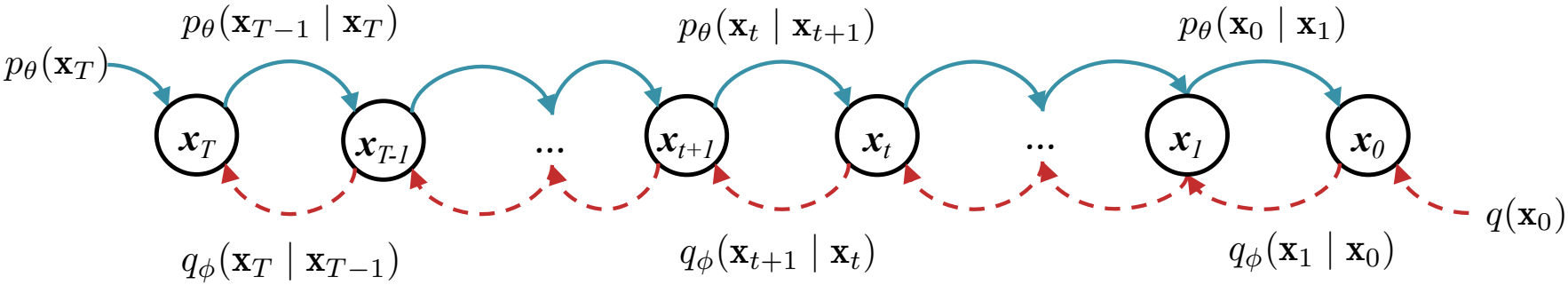
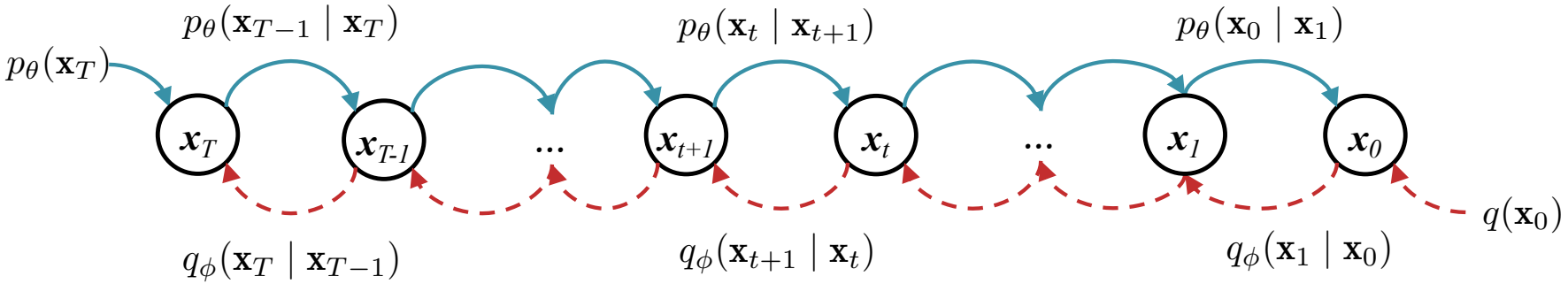


Figure from Ho et al. (2020)

Denoising Diffusion Probabilistic Model (DDPM)



Forward Process:

$$q_\phi(\mathbf{x}_{0:T}) = q(\mathbf{x}_0) \prod_{t=1}^{T} q_\phi(\mathbf{x}_t \mid \mathbf{x}_{t-1})$$

$$q(\mathbf{x}_0) = \text{data distribution}$$
$$q_\phi(\mathbf{x}_t \mid \mathbf{x}_{t-1}) \sim \mathcal{N}(\sqrt{\alpha_t}\mathbf{x}_{t-1}, (1-\alpha_t)\mathbf{I})$$

(Learned) Reverse Process:

$$p_\theta(\mathbf{x}_{0:T}) = p_\theta(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$$

$$p_\theta(\mathbf{x}_T) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$
$$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) \sim \mathcal{N}(\mu_\theta(\mathbf{x}_t, t), \mathbf{\Sigma}_\theta(\mathbf{x}_t, t))$$

Gaussian (an aside)

Let $X \sim \mathcal{N}(\mu_x, \sigma_x^2)$ and $Y \sim \mathcal{N}(\mu_y, \sigma_y^2)$

1. Sum of two Gaussians is a Gaussian

$$X + Y \sim \mathcal{N}(\mu_x + \mu_y, \sigma_x^2 + \sigma_y^2)$$

2. Difference of two Gaussians is a Gaussian

$$X - Y \sim \mathcal{N}(\mu_x - \mu_y, \sigma_x^2 + \sigma_y^2)$$

3. Gaussian with a Gaussian mean has a Gaussian Conditional

$$Z \sim \mathcal{N}(\mu_z = X, \sigma_z^2) \Rightarrow P(Z \mid X) \sim \mathcal{N}(\cdot, \cdot)$$

Defining the Forward Process

Forward Process:

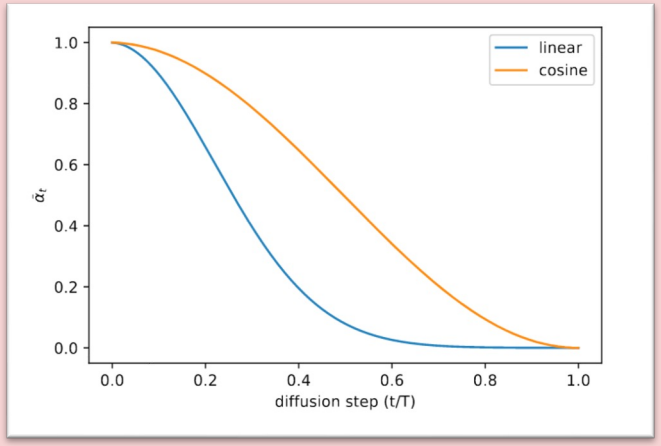
$$q_\phi(\mathbf{x}_{0:T}) = q(\mathbf{x}_0) \prod_{t=1}^{T} q_\phi(\mathbf{x}_t \mid \mathbf{x}_{t-1})$$

$$q(\mathbf{x}_0) = \text{data distribution}$$

$$q_\phi(\mathbf{x}_t \mid \mathbf{x}_{t-1}) \sim \mathcal{N}(\sqrt{\alpha_t}\mathbf{x}_{t-1}, (1-\alpha_t)\mathbf{I})$$

Noise schedule:

We choose α_t to follow a fixed schedule s.t. $q_\phi(\mathbf{x}_T) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, just like $p_\theta(\mathbf{x}_T)$.



Property #1:

$$q(\mathbf{x}_t \mid \mathbf{x}_0) \sim \mathcal{N}(\sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$$

where $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$

Q: So what is $q_\phi(\mathbf{x}_T \mid \mathbf{x}_0)$? Note the *capital* T in the subscript.

A:

$$q(\mathbf{x}_T \mid \mathbf{x}_0) \sim \mathcal{N}(\boldsymbol{\mu} \approx \mathbf{0}, \boldsymbol{\Sigma} \approx \mathbf{I})$$

Gaussian (an aside)

Let $X \sim \mathcal{N}(\mu_x, \sigma_x^2)$ and $Y \sim \mathcal{N}(\mu_y, \sigma_y^2)$

1. Sum of two Gaussians is a Gaussian

$$X + Y \sim \mathcal{N}(\mu_x + \mu_y, \sigma_x^2 + \sigma_y^2)$$

2. Difference of two Gaussians is a Gaussian

$$X - Y \sim \mathcal{N}(\mu_x - \mu_y, \sigma_x^2 + \sigma_y^2)$$

3. Gaussian with a Gaussian mean has a Gaussian Conditional

$$Z \sim \mathcal{N}(\mu_z = X, \sigma_z^2) \Rightarrow P(Z \mid X) \sim \mathcal{N}(\cdot, \cdot)$$

4. But #3 does not hold if X is passed through a nonlinear function f

$$W \sim \mathcal{N}(\mu_z = f(X), \sigma_w^2) \not\Rightarrow P(W \mid X) \sim \mathcal{N}(\cdot, \cdot)$$

Parameterizing the *learned* reverse process

Recall: $p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) \sim \mathcal{N}(\mu_\theta(\mathbf{x}_t, t), \mathbf{\Sigma}_\theta(\mathbf{x}_t, t))$

Later we will show that given a training sample $\mathbf{x}_0$, we want

$$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$$

to be as close as possible to

$$q(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0)$$

Intuitively, this makes sense: if the *learned* reverse process is supposed to subtract away the noise, then whenever we're working with a specific $\mathbf{x}_0$ it should subtract it away exactly as *exact* reverse process would have.

Properties of forward and *exact* reverse processes

Property #1:

$$q(\mathbf{x}_t \mid \mathbf{x}_0) \sim \mathcal{N}(\sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$$

$$\text{where } \bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$$

$\Rightarrow$ we can sample $\mathbf{x}_t$ from $\mathbf{x}_0$ at any timestep t efficiently in closed form

$\Rightarrow$ $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + (1-\bar{\alpha}_t)\boldsymbol{\epsilon}$ where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

Property #2: Estimating $q(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$ is intractable because of its dependence on $q(\mathbf{x}_0)$. However, conditioning on $\mathbf{x}_0$ we can efficiently work with:

$$q(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\tilde{\mu}_q(\mathbf{x}_t, \mathbf{x}_0), \sigma_t^2 \mathbf{I})$$

$$\begin{aligned}\text{where } \tilde{\mu}_q(\mathbf{x}_t, \mathbf{x}_0) &= \frac{\sqrt{\bar{\alpha}_t}(1-\alpha_t)}{1-\bar{\alpha}_t}\mathbf{x}_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_t)}{1-\bar{\alpha}_t}\mathbf{x}_t \\ &= \alpha_t^{(0)}\mathbf{x}_0 + \alpha_t^{(t)}\mathbf{x}_t \\ \sigma_t^2 &= \frac{(1-\bar{\alpha}_{t-1})(1-\alpha_t)}{1-\bar{\alpha}_t}\end{aligned}$$

Property #3: Combining the two previous properties, we can obtain a different parameterization of $\tilde{\mu}_q$ which has been shown empirically to help in learning p_θ.

Rearranging $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$ we have that:

$$\mathbf{x}_0 = \left(\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}\right) / \sqrt{\bar{\alpha}_t}$$

Substituting this definition of $\mathbf{x}_0$ into property #2's definition of $\tilde{\mu}_q$ gives:

$$\begin{aligned}\tilde{\mu}_q(\mathbf{x}_t, \mathbf{x}_0) &= \alpha_t^{(0)}\mathbf{x}_0 + \alpha_t^{(t)}\mathbf{x}_t \\ &= \alpha_t^{(0)}\left(\left(\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}\right) / \sqrt{\bar{\alpha}_t}\right) + \alpha_t^{(t)}\mathbf{x}_t \\ &= \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{(1-\alpha_t)}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}\right)\end{aligned}$$

Parameterizing the *learned* reverse process

Recall: $p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) \sim \mathcal{N}(\mu_\theta(\mathbf{x}_t, t), \mathbf{\Sigma}_\theta(\mathbf{x}_t, t))$

Idea #1: Rather than learn $\Sigma_\theta(\mathbf{x}_t, t)$ just use what we know about $q(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0) \sim \mathcal{N}(\tilde{\mu}_q(\mathbf{x}_t, \mathbf{x}_0), \sigma_t^2\mathbf{I})$:

$$\Sigma_\theta(\mathbf{x}_t, t) = \sigma_t^2\mathbf{I}$$

Idea #2: Choose μ_θ based on $q(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0)$, i.e. we want $\mu_\theta(\mathbf{x}_t, t)$ to be close to $\tilde{\mu}_q(\mathbf{x}_t, \mathbf{x}_0)$. Here are three ways we could parameterize this:

Option A: Learn a network that approximates $\tilde{\mu}_q(\mathbf{x}_t, \mathbf{x}_0)$ directly from $\mathbf{x}_t$ and t:

$$\mu_\theta(\mathbf{x}_t, t) = \text{UNet}_\theta(\mathbf{x}_t, t)$$

where t is treated as an extra feature in UNet

Option B: Learn a network that approximates the real $\mathbf{x}_0$ from only $\mathbf{x}_t$ and t:

$$\mu_\theta(\mathbf{x}_t, t) = \alpha_t^{(0)}\mathbf{x}_\theta^{(0)}(\mathbf{x}_t, t) + \alpha_t^{(t)}\mathbf{x}_t$$

where $\mathbf{x}_\theta^{(0)}(\mathbf{x}_t, t) = \text{UNet}_\theta(\mathbf{x}_t, t)$

Option C: Learn a network that approximates the ϵ that gave rise to $\mathbf{x}_t$ from $\mathbf{x}_0$ in the forward process from $\mathbf{x}_t$ and t:

$$\mu_\theta(\mathbf{x}_t, t) = \alpha_t^{(0)}\mathbf{x}_\theta^{(0)}(\mathbf{x}_t, t) + \alpha_t^{(t)}\mathbf{x}_t$$

where $\mathbf{x}_\theta^{(0)}(\mathbf{x}_t, t) = \left(\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right)/\sqrt{\bar{\alpha}_t}$

where $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) = \text{UNet}_\theta(\mathbf{x}_t, t)$

DIFFUSION MODEL TRAINING

Learning the Reverse Process

Recall: given a training sample $\mathbf{x}_0$, we want

$$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$$

to be as close as possible to

$$q(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0)$$

Depending on which of the options for parameterization we pick, we get a different training algorithm.

Algorithm 1 Training (Option A, all timesteps)

initialize θ
for $e \in \{1, \dots, E\}$ **do**
  for $x_0 \in \mathcal{D}$ **do**
    for $t \in \{1, \dots, T\}$ **do**
      $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
      $\mathbf{x}_t \leftarrow \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$
      $\tilde{\mu}_q \leftarrow \alpha_t^{(0)}\mathbf{x}_0 + \alpha_t^{(t)}\mathbf{x}_t$
      $\ell_t(\theta) \leftarrow \|\tilde{\mu}_q - \mu_\theta(\mathbf{x}_t, t)\|^2$
    $\theta \leftarrow \theta - \nabla_\theta \sum_{t=1}^{T} \ell_t(\theta)$

Option A: Learn a network that approximates $\tilde{\mu}_q(\mathbf{x}_t, \mathbf{x}_0)$ directly from $\mathbf{x}_t$ and t:

$$\mu_\theta(\mathbf{x}_t, t) = \text{UNet}_\theta(\mathbf{x}_t, t)$$

where t is treated as an extra feature in UNet

Learning the Reverse Process

Recall: given a training sample $\mathbf{x}_0$, we want

$$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$$

to be as close as possible to

$$q(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0)$$

Depending on which of the options for parameterization we pick, we get a different training algorithm.

Algorithm 1 Training (Option A)

1: initialize θ
2: **for** $e \in \{1, \ldots, E\}$ **do**
3: **for** $x_0 \in \mathcal{D}$ **do**
4: $t \sim \text{Uniform}(1, \ldots, T)$
5: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
6: $\mathbf{x}_t \leftarrow \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$
7: $\tilde{\mu}_q \leftarrow \alpha_t^{(0)}\mathbf{x}_0 + \alpha_t^{(t)}\mathbf{x}_t$
8: $\ell_t(\theta) \leftarrow \|\tilde{\mu}_q - \mu_\theta(\mathbf{x}_t, t)\|^2$
9: $\theta \leftarrow \theta - \nabla_\theta \ell_t(\theta)$

Option A: Learn a network that approximates $\tilde{\mu}_q(\mathbf{x}_t, \mathbf{x}_0)$ directly from $\mathbf{x}_t$ and t:

$$\mu_\theta(\mathbf{x}_t, t) = \text{UNet}_\theta(\mathbf{x}_t, t)$$

where t is treated as an extra feature in UNet

Learning the Reverse Process

Recall: given a training sample $\mathbf{x}_0$, we want

$$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$$

to be as close as possible to

$$q(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0)$$

Depending on which of the options for parameterization we pick, we get a different training algorithm.

Algorithm 1 Training (Option B)

1: initialize θ
2: **for** $e \in \{1, \dots, E\}$ **do**
3: **for** $x_0 \in \mathcal{D}$ **do**
4: $t \sim \text{Uniform}(1, \dots, T)$
5: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
6: $\mathbf{x}_t \leftarrow \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t}\boldsymbol{\epsilon}$
7: $\ell_t(\theta) \leftarrow \|\mathbf{x}_0 - \mathbf{x}_\theta^{(0)}(\mathbf{x}_t, t)\|^2$
8: $\theta \leftarrow \theta - \nabla_\theta \ell_t(\theta)$

Option B: Learn a network that approximates the real $\mathbf{x}_0$ from only $\mathbf{x}_t$ and t:

$$\mu_\theta(\mathbf{x}_t, t) = \alpha_t^{(0)}\mathbf{x}_\theta^{(0)}(\mathbf{x}_t, t) + \alpha_t^{(t)}\mathbf{x}_t$$

where $\mathbf{x}_\theta^{(0)}(\mathbf{x}_t, t) = \text{UNet}_\theta(\mathbf{x}_t, t)$

Learning the Reverse Process

Recall: given a training sample $\mathbf{x}_0$, we want

$$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$$

to be as close as possible to

$$q(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0)$$

Depending on which of the options for parameterization we pick, we get a different training algorithm.

Option C is the best empirically

Algorithm 1 Training (Option C)

1: initialize θ
2: **for** $e \in \{1, \dots, E\}$ **do**
3: **for** $x_0 \in \mathcal{D}$ **do**
4: $t \sim \text{Uniform}(1, \dots, T)$
5: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
6: $\mathbf{x}_t \leftarrow \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$
7: $\ell_t(\theta) \leftarrow \|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\|^2$
8: $\theta \leftarrow \theta - \nabla_\theta \ell_t(\theta)$

Option C: Learn a network that approximates the ϵ that gave rise to $\mathbf{x}_t$ from $\mathbf{x}_0$ in the forward process from $\mathbf{x}_t$ and t:

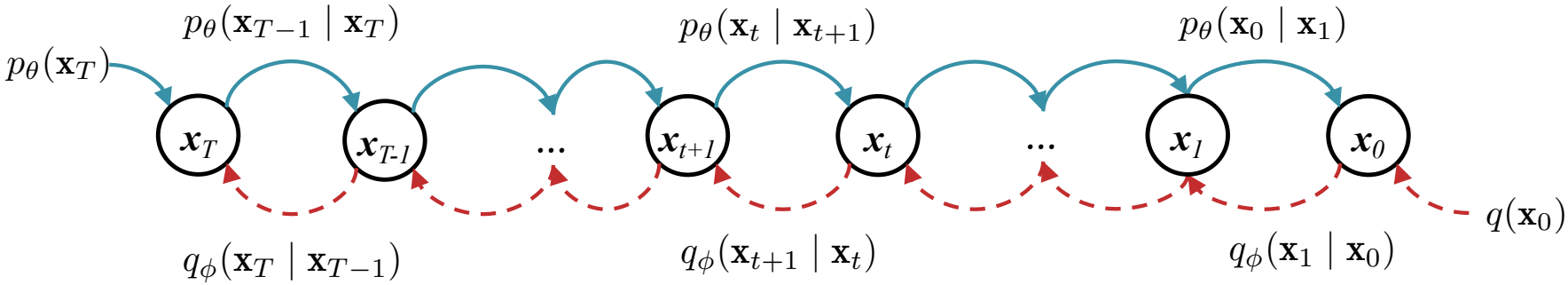
$$\mu_\theta(\mathbf{x}_t, t) = \alpha_t^{(0)}\mathbf{x}_\theta^{(0)}(\mathbf{x}_t, t) + \alpha_t^{(t)}\mathbf{x}_t$$

where $\mathbf{x}_\theta^{(0)}(\mathbf{x}_t, t) = \left(\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right)/\sqrt{\bar{\alpha}_t}$

where $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) = \text{UNet}_\theta(\mathbf{x}_t, t)$

Training (Computation Graph)

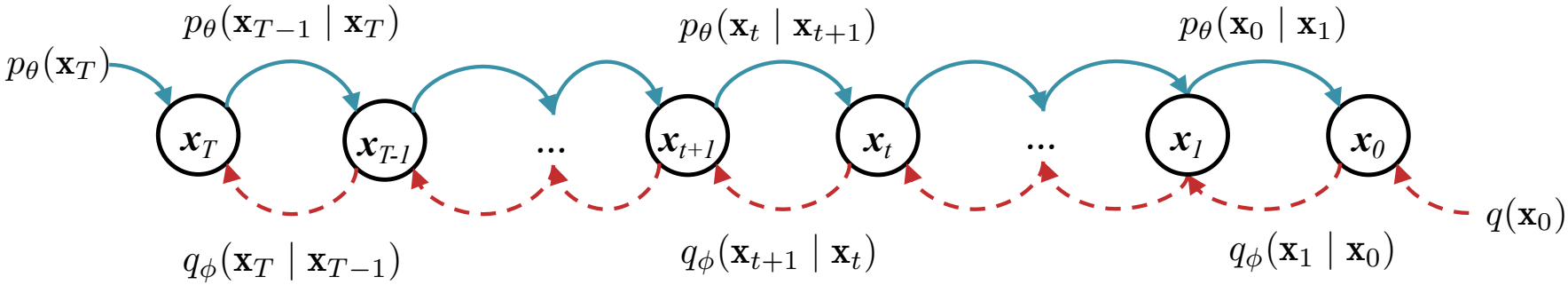
Sampling from the *learned* reverse process



Algorithm 1 Sampling

1: $\mathbf{x}_T \sim p_\theta(\mathbf{x}_T)$
2: **for** $t \in \{T, \dots, 1\}$ **do**
3: $\quad \mathbf{x}_{t-1} \sim p(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$
4: **return** $\mathbf{x}_0$

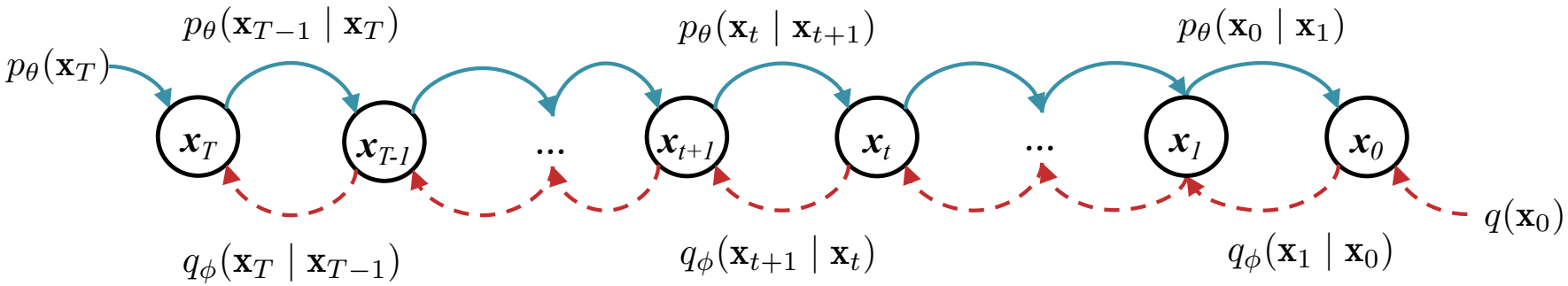
Sampling from the *learned* reverse process



Algorithm 1 Sampling

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t \in \{T, \dots, 1\}$ **do**
3: $\quad \mathbf{x}_{t-1} \sim \mathcal{N}(\mu_\theta(\mathbf{x}_t, t), \mathbf{\Sigma}_\theta(\mathbf{x}_t, t))$
4: **return** $\mathbf{x}_0$

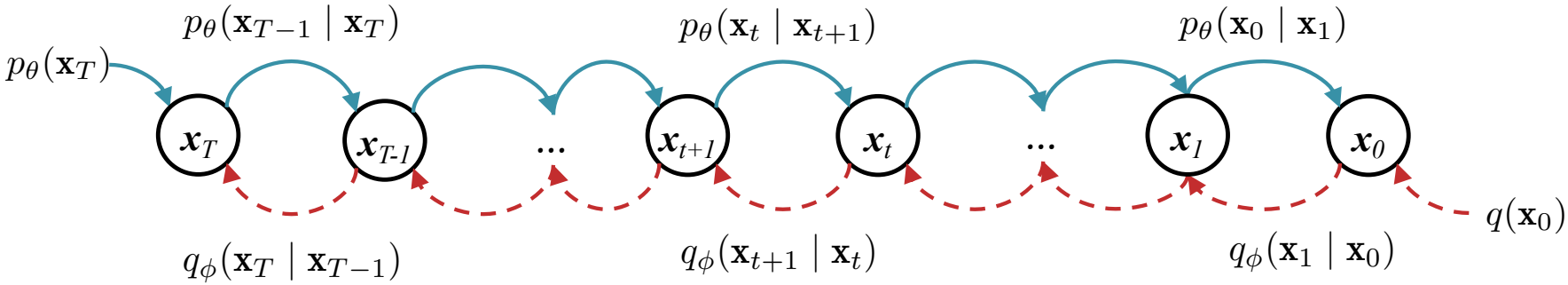
Sampling from the *learned* reverse process



Algorithm 1 Sampling (Option A)

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t \in \{T, \dots, 1\}$ **do**
3: $\quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
4: $\quad \mathbf{x}_{t-1} \leftarrow \mu_\theta(\mathbf{x}_t, t) + \sigma_t^2 \boldsymbol{\epsilon}$
5: **return** $\mathbf{x}_0$

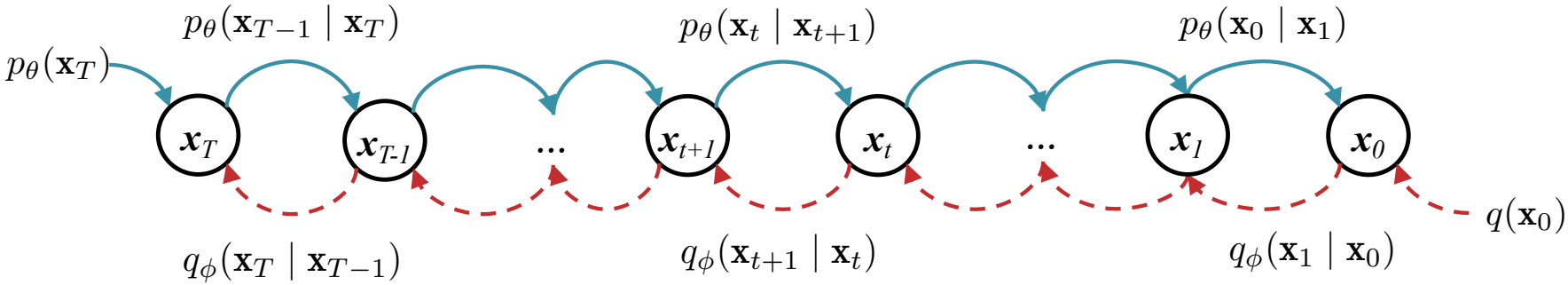
Sampling from the *learned* reverse process



Algorithm 1 Sampling (Option B)

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t \in \{T, \dots, 1\}$ **do**
3: $\quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
4: $\quad \hat{\boldsymbol{\mu}}_t \leftarrow \alpha_t^{(0)} \mathbf{x}_\theta^{(0)}(\mathbf{x}_t, t) + \alpha_t^{(t)} \mathbf{x}_t$
5: $\quad \mathbf{x}_{t-1} \leftarrow \hat{\boldsymbol{\mu}}_t + \sigma_t^2 \boldsymbol{\epsilon}$
6: **return** $\mathbf{x}_0$

Sampling from the *learned* reverse process



Algorithm 1 Sampling (Option C)

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t \in \{T, \dots, 1\}$ **do**
3: $\quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
4: $\quad \hat{\mathbf{x}}_0 \leftarrow \left(\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right) / \sqrt{\bar{\alpha}_t}$
5: $\quad \hat{\boldsymbol{\mu}}_t \leftarrow \alpha_t^{(0)}\hat{\mathbf{x}}_0 + \alpha_t^{(t)}\mathbf{x}_t$
6: $\quad \mathbf{x}_{t-1} \leftarrow \hat{\boldsymbol{\mu}}_t + \sigma_t^2\boldsymbol{\epsilon}$
7: **return** $\mathbf{x}_0$

Unsupervised Learning

Assumptions:

1. our data comes from some distribution $p^*(\mathbf{x}_0)$
2. we choose a distribution $p_\theta(\mathbf{x}_0)$ for which sampling $x_0 \sim p_\theta(\mathbf{x}_0)$ is tractable

Goal: learn θ s.t. $p_\theta(\mathbf{x}_0) \approx p^*(\mathbf{x}_0)$

Example: Diffusion Models

- true $p^*(\mathbf{x}_0)$ is distribution over photos taken and posted to Flikr
- choose $p_\theta(\mathbf{x}_0)$ to be an expressive model (e.g. noise fed into inverted CNN) that can generate images
  - sampling is will be easy
- learn by finding $\theta \approx \text{argmax}_\theta \log(p_\theta(\mathbf{x}_0))$?
  - Sort of! We can't compute the gradient $\nabla_\theta \log(p_\theta(\mathbf{x}_0))$
  - So we instead optimize a variational lower bound (more on that later)

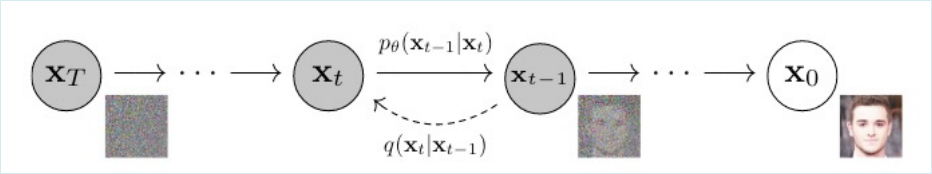


Figure from Ho et al. (2020)

DDPM Objective Function

$$\mathbb{E}\left[-\log p_\theta(\mathbf{x}_0)\right] \leq \mathbb{E}_q\left[-\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\right] = \mathbb{E}_q\left[-\log p(\mathbf{x}_T) - \sum_{t\geq 1} \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_t|\mathbf{x}_{t-1})}\right] =: L$$

$$L = \mathbb{E}_q\Bigg[\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_T|\mathbf{x}_0) \parallel p(\mathbf{x}_T))}_{L_T} + \sum_{t>1} \underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{L_{t-1}} \underbrace{-\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)}_{L_0}\Bigg]$$

This KL divergence term L_{t-1} wants the two conditional distributions to be as close as possible.

Equations from Ho et al. (2020)

Connections between VAE and Diffusion

Variational Autoencoder

VAE The VAE learns via stochastic gradient variational Bayes (SGVB):

- true posterior: $p_\theta(\mathbf{z} \mid \mathbf{x})$ (intractable)
- variational approx: $q_\phi(\mathbf{z} \mid \mathbf{x})$ (encoder)
- model: $p_\theta(\mathbf{x}, \mathbf{z}) = p_\theta(\mathbf{x} \mid \mathbf{z}) p_{\phi)(\mathbf{z})}$
  where $p_\theta(\mathbf{x} \mid \mathbf{z})$ (decoder) and $p_\theta(\mathbf{z})$ (Gaussian)
- learning samples an example i:

$$\widetilde{\mathcal{L}}^B(\theta, \phi; \mathbf{x}^{(i)}) = -D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}^{(i)}) \| p_\theta(\mathbf{z})) + \log p_\theta(\mathbf{x}^{(i)}|\mathbf{z}^{(i)})$$
$$[\theta, \phi] \leftarrow [\theta, \phi] - \nabla_{[\theta,\phi]} \widetilde{\mathcal{L}}^B(\theta, \phi; \mathbf{x}^{(i)})$$

Denoising Diffusion Probabilistic Model

DDPM The DDPM learns in a similar fashion:

- true posterior: $p_\theta(\mathbf{x}_1, \ldots, \mathbf{x}_T \mid \mathbf{x}_0)$ (intractable)
- variational approx: $q_\phi(\mathbf{x}_{0:T})$ (forward process)
- model: $p_\theta(\mathbf{x}_{0:T})$ (reverse process)
- learning only learns θ (i.e. no inference):

$$\ell_t(\theta) \leftarrow \|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\|^2$$
$$\theta \leftarrow \theta - \nabla_\theta \ell_t(\theta)$$

Zero-shot Learning

- Task: at test time, make predictions about labels or classes *that were not observed during training*
  - Relevant in settings with many, potentially rare classes or where new classes arise over time

$z \rightarrow$	1	2	3	4	5
x_t	y_t^1	y_t^2	y_t^3	y_t^4	y_t^5
	1	0	0	-	-
	0	1	0	-	-
	0	0	1	-	-
	-	-	-	1	0
	-	-	-	0	1
	training data			test data	

- How can we possibly do well in this setting???
- Idea: cheat!

Source: https://cdn.aaai.org/AAAI/2008/AAAI08-103.pdf

Zero-shot Learning

- Task: at test time, make predictions about labels or classes *that were not observed during training*
  - Relevant in settings with many, potentially rare classes or where new classes arise over time

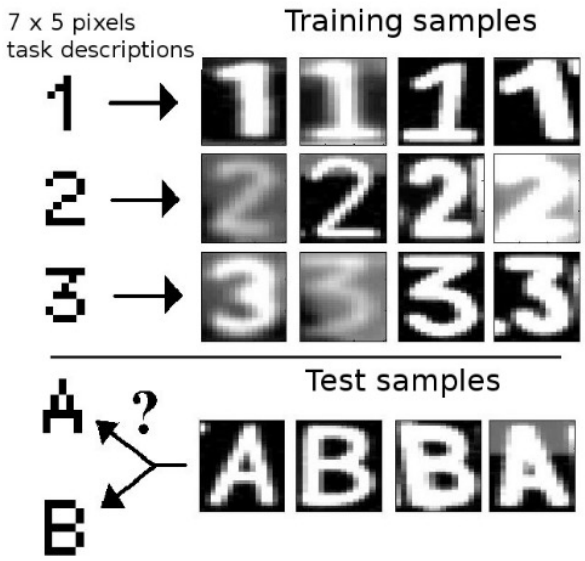
$z \rightarrow$ $d(z) \rightarrow$	1 	2 	3 	4 	5 
x_t	y_t^1	y_t^2	y_t^3	y_t^4	y_t^5
	1	0	0	-	-
	0	1	0	-	-
	0	0	1	-	-
	-	-	-	1	0
	-	-	-	0	1
	training data			test data	

- How can we possibly do well in this setting???
- Idea: ~~cheat~~! leverage a "description of the label", $d(z)$

Source: https://cdn.aaai.org/AAAI/2008/AAAI08-103.pdf

Zero-shot Learning

- Task: at test time, make predictions about labels or classes *that were not observed during training*
  - Relevant in settings with many, potentially rare classes or where new classes arise over time



- Train a score function on (input, description) pairs $f\left(x^{(i)}, d\left(z^{(j)}\right)\right)$ that is high for the true label's description and low otherwise.
- At test time, predict the label with the highest scoring description

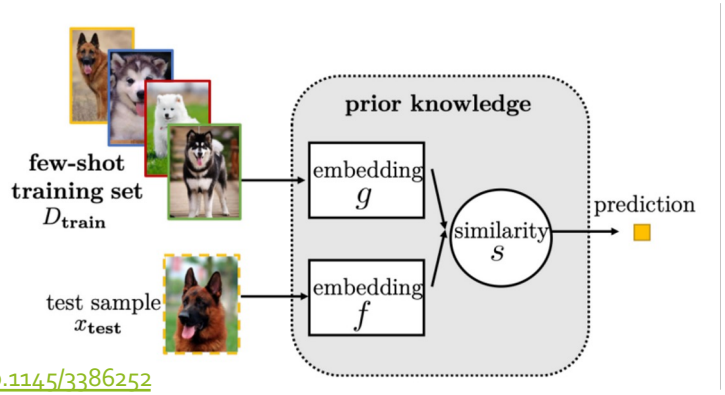
Source: https://cdn.aaai.org/AAAI/2008/AAAI08-103.pdf

Zero-shot Learning

- Task: at test time, make predictions about labels or classes *that were not observed during training*
  - Relevant in settings with many, potentially rare classes or where new classes arise over time
  - Traditional zero-shot learning methods typically require access to "semantic" or "auxiliary" information *about both seen and unseen classes* e.g., manually-defined attributes or raw text descriptions

Few-shot Learning

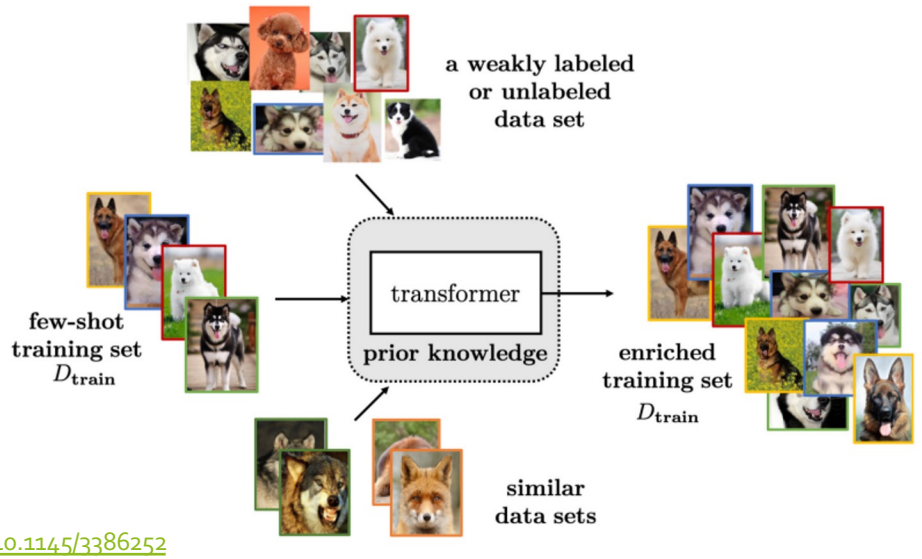
- Task: at test time, make predictions about labels or classes *that were barely observed during training*
  - Typically refers to settings with fewer than 5 instances of each class
  - Idea: learn a low-dimensional embedding for the inputs and do some kNN style prediction in the latent space



Source: https://dl.acm.org/doi/10.1145/3386252

Few-shot Learning

- Task: at test time, make predictions about labels or classes *that were barely observed during training*
  - Typically refers to settings with fewer than 5 instances of each class
  - Idea: data augmentation = create more (potentially bad, potentially irrelevant, potentially wrong) training inputs



Source: https://dl.acm.org/doi/10.1145/3386252

Few-shot Learning

- Task: at test time, make predictions about labels or classes *that were barely observed during training*
  - Typically refers to settings with fewer than 5 instances of each class
  - Idea: just throw a ~~neural network~~ LLM at it?

Zero-shot Learning with LLMs

- Recall: language models are trained to maximize the likelihood of sequences in the training data
- Key idea: provide a prefix sequence or "prompt" to the (large) language model such that its likely completion is the answer you want!
  - Can do conditional sampling from the model's learned distribution over next tokens to generate completions:

$$p(x_{T+1} \mid \text{PROMPT} = [x_1, \ldots, x_T])$$

Zero-shot Learning with LLMs: Example

- Question answering ("entailment")

Prompt:

Rhodochiton is a genus of flowering plants within the family Plantaginaceae, native to southern Mexico and neighbouring Guatemala. They climb by means of twining leaf stalks. One of the three species, "Rhodochiton atrosanguineus", the purple bell vine, is grown as an ornamental plant. All three species are sometimes included in "Lophospermum".

Question: You can find the purple bell vine in more than one country. True, False, or Neither?

Model output (Llama-2-70B):

Answer: True.

Explanation: The purple bell vine (Rhodochiton atrosanguineus) is native to southern Mexico and neighbouring Guatemala, which means it can be found in more than one country.

Correct answer:

True.

Source: https://huggingface.co/datasets/facebook/anli

Zero-shot Learning with LLMs: Example

- Summarization

Prompt:

Story: Jason listened to the weather and heard it was going to be sunny. He thought the kids might like to go swimming. He gathered up the swimsuits, towels and sunscreen. Jason and the kids got into the truck and drove to the beach. They spent the next 2 hours playing and splashing in the surf.

One-sentence Summary:

Model output (Llama-2-70B):

Jason took the kids swimming at the beach after hearing the weather forecast, gathering necessary items and driving there.

Source: https://arxiv.org/pdf/1910.00998.pdf

Zero-shot Learning with LLMs: Example

- Machine translation

Prompt:

El último dueño de esta propiedad había sido un hombre soltero, que alcanzó una muy avanzada edad, y que durante gran parte de su existencia tuvo en su hermana una fiel compañera y ama de casa.

English translation:

Model output (Llama-2-70B):

The last owner of this property was a bachelor man who lived to a very advanced age, and who had his sister as a faithful companion and housekeeper for most of his life.

Correct answer (from Sense & Sensibility):

The late owner of this estate was a single man, who lived to a very advanced age, and who for many years of his life, had a constant companion and housekeeper in his sister.

Source: https://huggingface.co/datasets/opus_books/viewer/en-es

How is this possible? Why does this work?

- Machine translation

Prompt:

El último dueño de esta propiedad había sido un hombre soltero, que alcanzó una muy avanzada edad, y que durante gran parte de su existencia tuvo en su hermana una fiel compañera y ama de casa.

English translation:

Model output (Llama-2-70B):

The last owner of this property was a bachelor man who lived to a very advanced age, and who had his sister as a faithful companion and housekeeper for most of his life.

Correct answer (from Sense & Sensibility):

The late owner of this estate was a single man, who lived to a very advanced age, and who for many years of his life, had a constant companion and housekeeper in his sister.

Source: https://huggingface.co/datasets/opus_books/viewer/en-es

Zero-shot Learning with LLMs: Intuition

- Many NLP tasks appear (in some form) on the internet/in the training datasets for these LLMs

"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**".

Table 1. Examples of naturally occurring demonstrations of English to French and French to English translation found throughout the WebText training set.

Source: https://insightcivic.s3.us-east-1.amazonaws.com/language-models.pdf

Zero-shot Learning with LLMs: Intuition

- Many NLP tasks appear (in some form) on the internet/in the training datasets for these LLMs
- Zero-shot performance is highly dependent on model size

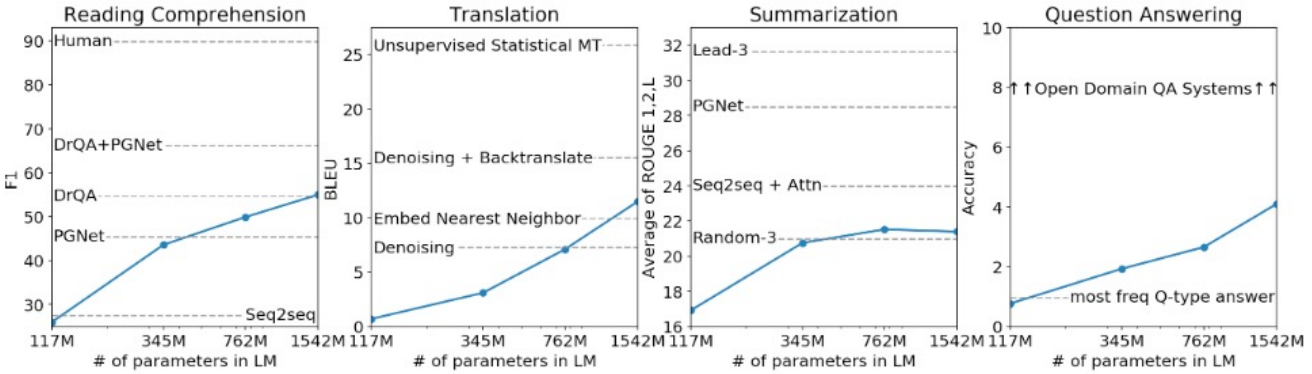


Figure 1. Zero-shot task performance of WebText LMs as a function of model size on many NLP tasks. Reading Comprehension results are on CoQA (Reddy et al., 2018), translation on WMT-14 Fr-En (Artetxe et al., 2017), summarization on CNN and Daily Mail (See et al., 2017), and Question Answering on Natural Questions (Kwiatkowski et al., 2019). Section 3 contains detailed descriptions of each result.

Source: https://insightcivic.s3.us-east-1.amazonaws.com/language-models.pdf

Zero-shot Learning with LLMs: Intuition

- Many NLP tasks appear (in some form) on the internet/in the training datasets for these LLMs
- Zero-shot performance is the result of ~~cheating~~ data contamination?

	PTB	WikiText-2	enwik8	text8	Wikitext-103	1BW
Dataset train	**2.67%**	0.66%	**7.50%**	2.34%	**9.09%**	**13.19%**
WebText train	0.88%	**1.63%**	6.31%	**3.94%**	2.42%	3.75%

Overall, our analysis suggests that data overlap between WebText training data and specific evaluation datasets provides a small but consistent benefit to reported results. However, for most datasets we do not notice significantly larger overlaps than those already existing between standard training and test sets, as Table 6 highlights.

Source: https://insightcivic.s3.us-east-1.amazonaws.com/language-models.pdf

Zero-shot Learning with LLMs: Intuition

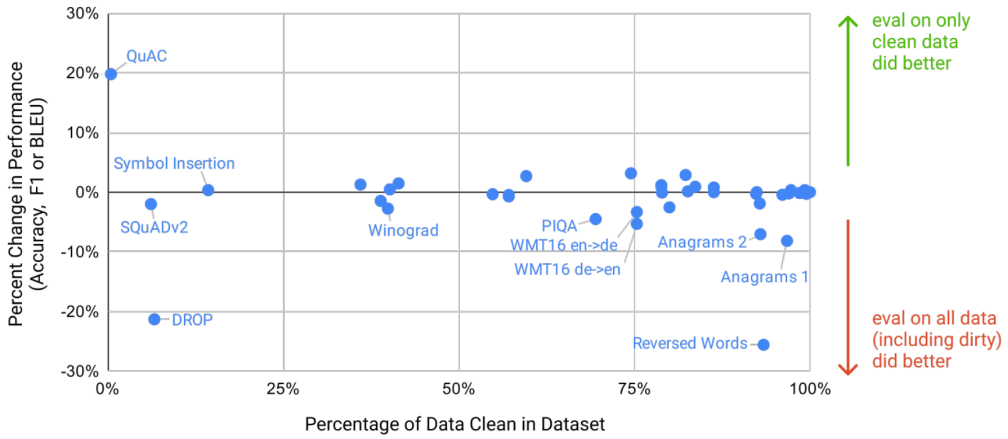
- Many NLP tasks appear (in some form) on the internet/in the training datasets for these LLMs
- Zero-shot performance is the result of ~~cheating~~ data contamination?

GPT-3 operates in a somewhat different regime. On the one hand, the dataset and model size are about two orders of magnitude larger than those used for GPT-2, and include a large amount of Common Crawl, creating increased potential for contamination and memorization. On the other hand, precisely due to the large amount of data, even GPT-3 175B does not overfit its training set by a significant amount, measured relative to a held-out validation set with which it was deduplicated (Figure 4.1). Thus, we expect that contamination is likely to be frequent, but that its effects may not be as large as feared.

Source: https://arxiv.org/pdf/2005.14165

Zero-shot Learning with LLMs: Intuition

- Many NLP tasks appear (in some form) on the internet/in the training datasets for these LLMs
- Zero-shot performance is the result of ~~cheating~~ data contamination?



- **Reading Comprehension:** Our initial analysis flagged >90% of task examples from QuAC, SQuAD2, and DROP as potentially contaminated, so large that even measuring the differential on a clean subset was difficult. Upon manual inspection, however, we found that for every overlap we inspected, in all 3 datasets, the source text was present in our training data but the question/answer pairs were not, meaning the model gains only background information and cannot memorize the answer to a specific question.

Source: https://arxiv.org/pdf/2005.14165

Zero-shot Learning with LLMs: Examples

- Answer fact-based questions:

Context →	Organisms require energy in order to do what?
Correct Answer →	mature and develop.
Incorrect Answer →	rest soundly.
Incorrect Answer →	absorb light.
Incorrect Answer →	take in nutrients.

- Complete sentences logically:

Context →	My body cast a shadow over the grass because
Correct Answer →	the sun was rising.
Incorrect Answer →	the grass was cut.

- Complete analogies:

Context →	lull is to trust as
Correct Answer →	cajole is to compliance
Incorrect Answer →	balk is to fortitude
Incorrect Answer →	betray is to loyalty
Incorrect Answer →	hinder is to destination
Incorrect Answer →	soothe is to passion

- Reading comprehension:

Context →	anli 1: anli 1: Fulton James MacGregor MSP is a Scottish politician who is a Scottish National Party (SNP) Member of Scottish Parliament for the constituency of Coatbridge and Chryston. MacGregor is currently Parliamentary Liaison Officer to Shona Robison, Cabinet Secretary for Health & Sport. He also serves on the Justice and Education & Skills committees in the Scottish Parliament. Question: Fulton James MacGregor is a Scottish politican who is a Liaison officer to Shona Robison who he swears is his best friend. True, False, or Neither?
Correct Answer →	Neither
Incorrect Answer →	True
Incorrect Answer →	False

Source: https://arxiv.org/pdf/2005.14165

Okay so clearly there's lots of examples of these: why are we doing zero-shot learning in the first place?

- Answer fact-based questions:

Context →	Organisms require energy in order to do what?
Correct Answer →	mature and develop.
Incorrect Answer →	rest soundly.
Incorrect Answer →	absorb light.
Incorrect Answer →	take in nutrients.

- Complete sentences logically:

Context →	My body cast a shadow over the grass because
Correct Answer →	the sun was rising.
Incorrect Answer →	the grass was cut.

- Complete analogies:

Context →	lull is to trust as
Correct Answer →	cajole is to compliance
Incorrect Answer →	balk is to fortitude
Incorrect Answer →	betray is to loyalty
Incorrect Answer →	hinder is to destination
Incorrect Answer →	soothe is to passion

- Reading comprehension:

Context →	anli 1: anli 1: Fulton James MacGregor MSP is a Scottish politician who is a Scottish National Party (SNP) Member of Scottish Parliament for the constituency of Coatbridge and Chryston. MacGregor is currently Parliamentary Liaison Officer to Shona Robison, Cabinet Secretary for Health & Sport. He also serves on the Justice and Education & Skills committees in the Scottish Parliament. Question: Fulton James MacGregor is a Scottish politican who is a Liaison officer to Shona Robison who he swears is his best friend. True, False, or Neither?
Correct Answer →	Neither
Incorrect Answer →	True
Incorrect Answer →	False

Source: https://arxiv.org/pdf/2005.14165

Few-shot Learning with LLMs

- Suppose you have…
  1. a small labelled dataset (i.e., a few-shot setting), $\mathcal{D}$
  2. a very large pre-trained language model
- There are two ways to "learn":

A. Supervised fine-tuning i.e., updating the LLM's parameters using
   1. a standard supervised objective
   2. backpropagation to compute gradients
   3. your favorite optimizer (e.g., Adam)

- **Con**: backpropagation requires ~3x the memory and computation time as the forward computation
- **Con**: you might not have access to the model parameters

Few-shot Learning with LLMs

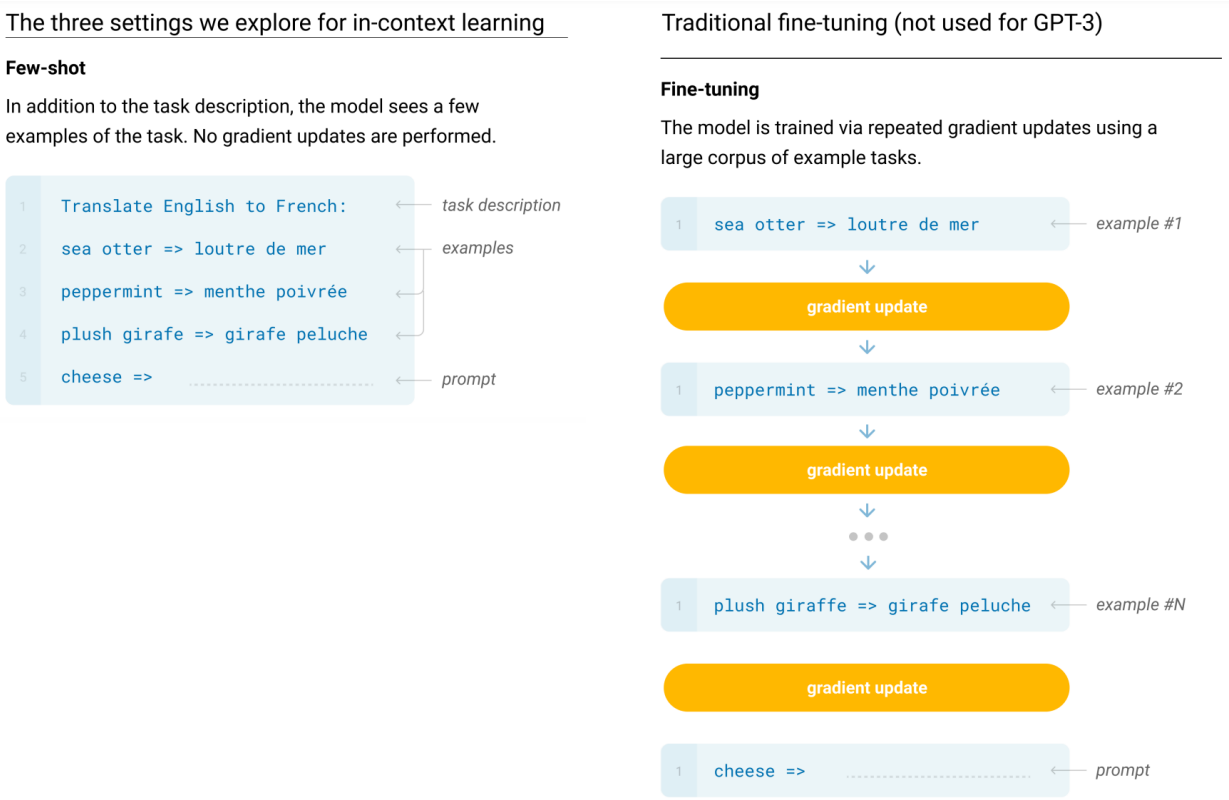
- Suppose you have…
  1. a small labelled dataset (i.e., a few-shot setting), $\mathcal{D}$
  2. a very large pre-trained language model
- There are two ways to "learn":

B. In-context learning i.e., feeding the training dataset to the LLM as a prompt and taking the output as a prediction
   - the LLM (hopefully) infers patterns in the training dataset during inference (i.e., decoding)

- **Pro**: no backpropagation required and only one pass through the training dataset *per test example*
- **Pro**: does not require access to the model parameters, only API access to the model itself
- **Con**: the prompt may be very long and Transformer LMs require $O(N^2)$ time/space where N = length of context

Few-shot Learning via In-context Learning with LLMs

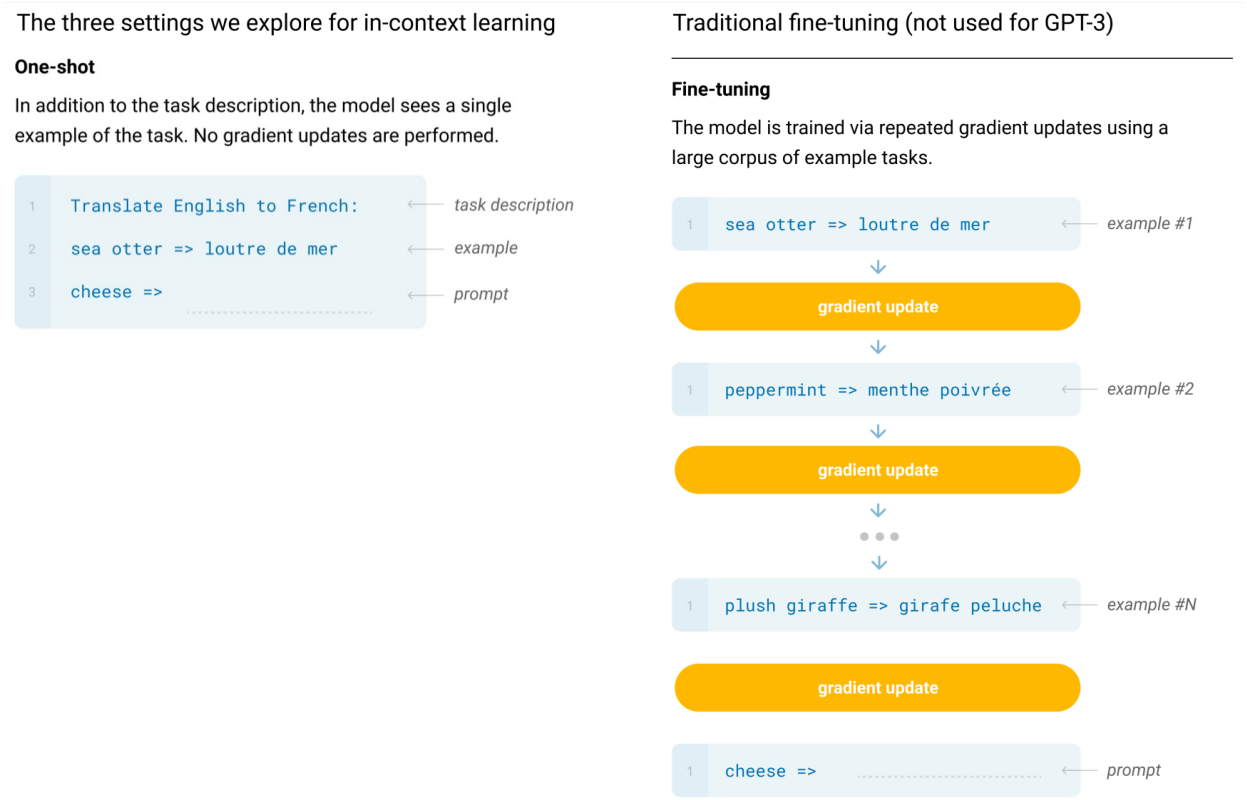
- Standard setup: a set of input/output pairs from a training dataset are presented in sequence to an LLM, typically along with a plain-text task description



Source: https://arxiv.org/pdf/2005.14165

Few-shot Learning via In-context Learning with LLMs

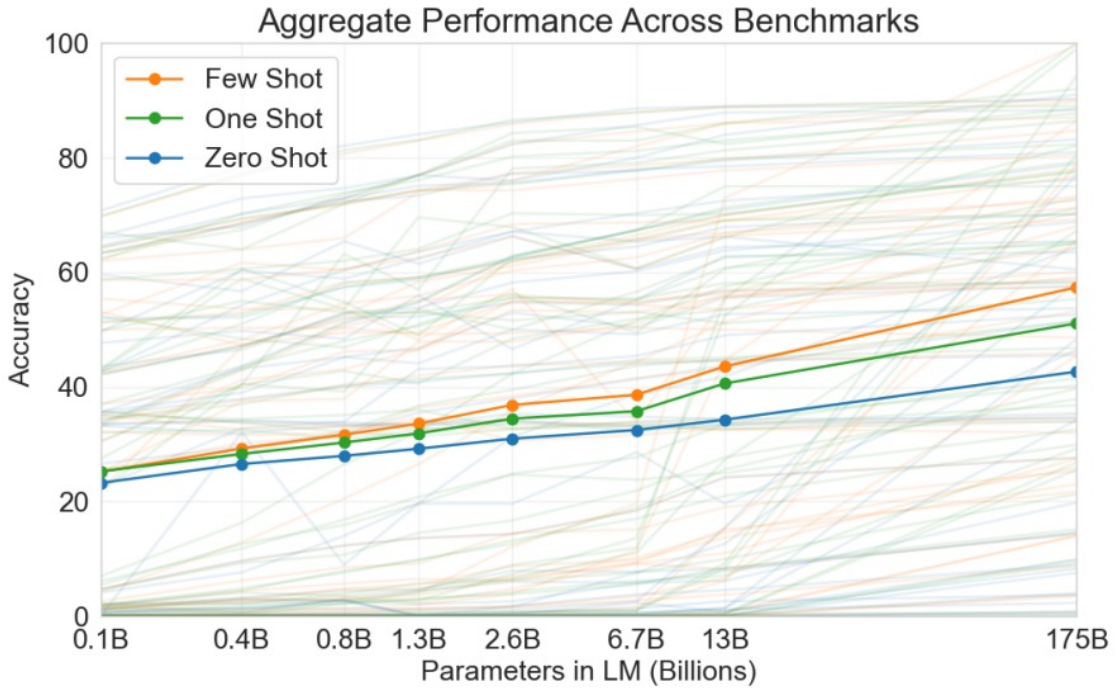
- Standard setup: a set of input/output pairs from a training dataset are presented in sequence to an LLM, typically along with a plain-text task description



Source: https://arxiv.org/pdf/2005.14165

Few-shot Learning via In-context Learning with LLMs

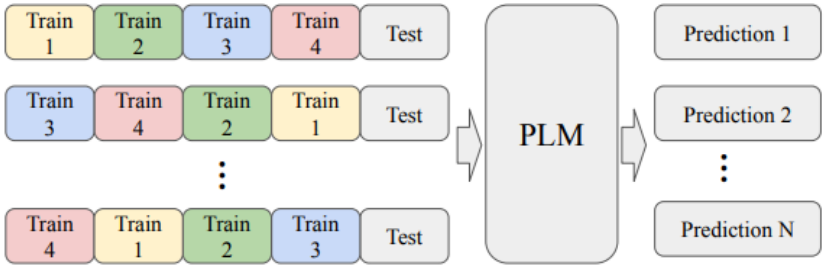
- Standard setup: a set of input/output pairs from a training dataset are presented in sequence to an LLM, typically along with a plain-text task description



Source: https://arxiv.org/pdf/2005.14165

Few-shot In-context Learning with LLMs

- In-context learning is surprisingly *sensitive* to…
  1. **the order the training examples are presented in**



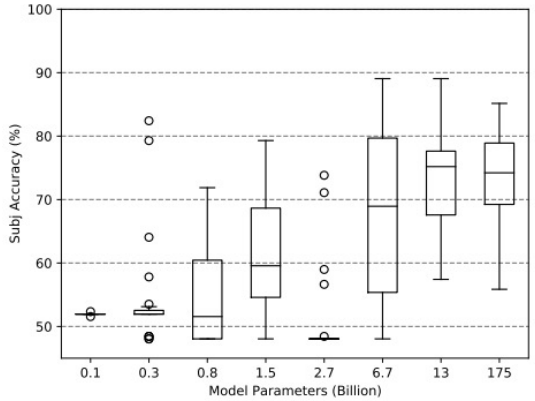
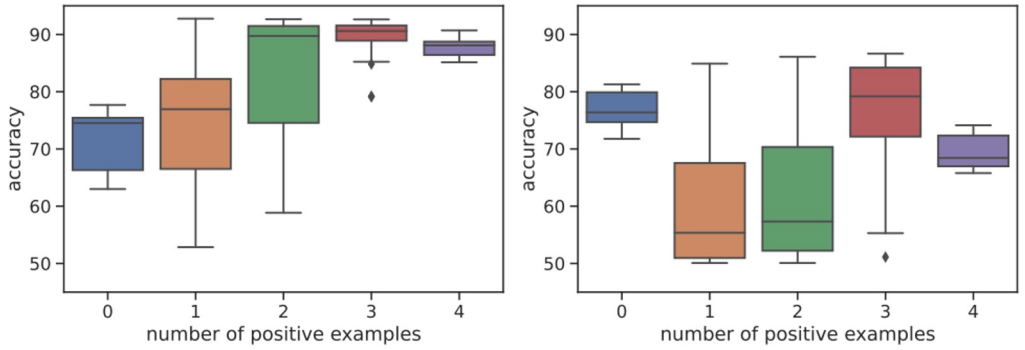


Figure 1: Four-shot performance for 24 different sample orders across different sizes of GPT-family models (GPT-2 and GPT-3) for the SST-2 and Subj datasets.

Source: http://arxiv.org/abs/2104.08786

Few-shot In-context Learning with LLMs

- In-context learning is surprisingly *sensitive* to…
  1. the order the training examples are presented in
  2. **label imbalance (e.g. # of positive vs. # of negative)**



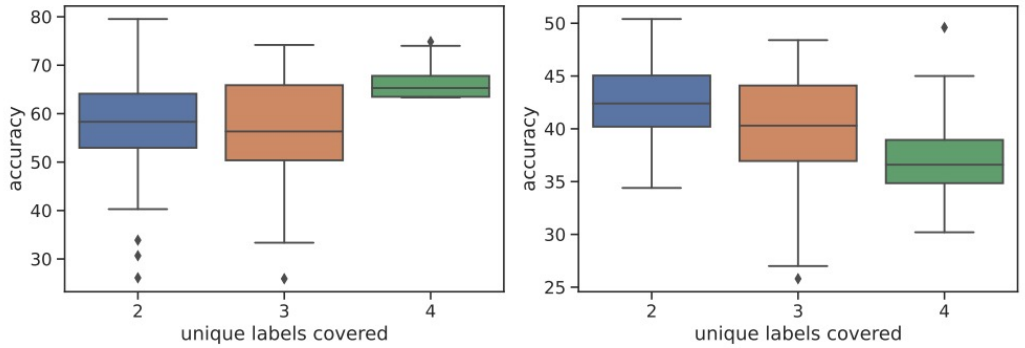
(a) Amazon (b) SST-2

Figure 3: Accuracies of Amazon and SST-2 with varying **label balance** (number of positive examples in demonstration), across 100 total random samples of 4 demonstration examples.

Source: https://aclanthology.org/2022.emnlp-main.622.pdf

Few-shot In-context Learning with LLMs

- In-context learning is surprisingly *sensitive* to…
  1. the order the training examples are presented in
  2. label imbalance (e.g. # of positive vs. # of negative)
  3. **the number of unique labels in the training dataset**



(a) AGNews (b) TREC

Figure 4: Accuracies of AGNews and TREC with varying **label coverage** (number of unique labels covered in demonstration), across 100 total random samples of 4 demonstration examples. Demonstration set that only covers 1 label is very unlikely and does not appear in our experiments.

Source: https://aclanthology.org/2022.emnlp-main.622.pdf

Few-shot In-context Learning with LLMs

- In-context learning is surprisingly *insensitive* to…
  1. **the correctness of the labels!**

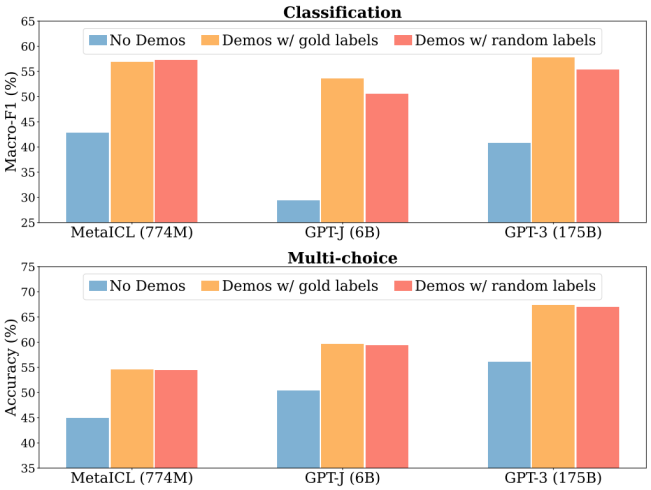


Figure 1: Results in classification (top) and multi-choice tasks (bottom), using three LMs with varying size. Reported on six datasets on which GPT-3 is evaluated; the channel method is used. See Section 4 for the full results. In-context learning performance drops only marginally when labels in the demonstrations are replaced by random labels.

Source: http://arxiv.org/abs/2202.12837

Few-shot In-context Learning with LLMs

- In-context learning is surprisingly *insensitive* to…
  1. the correctness of the labels!
  2. **the amount of training data used in the prompt!**

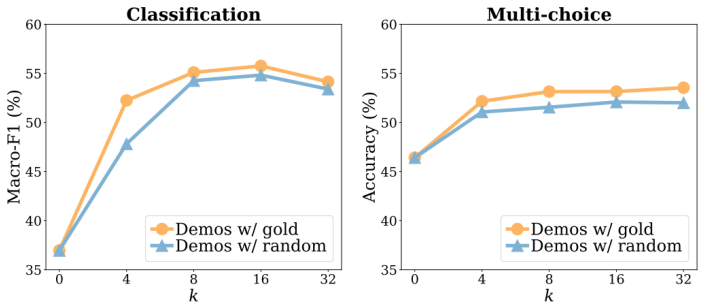


Figure 5: Ablations on varying numbers of examples in the demonstrations (k). Models that are the best under 13B in each task category (Channel MetaICL and Direct GPT-J, respectively) are used.

Source: http://arxiv.org/abs/2202.12837

So why does *this* work? Why is it better than zero-shot learning?

- In-context learning is surprisingly *insensitive* to…
  1. the correctness of the labels!
  2. **the amount of training data used in the prompt!**

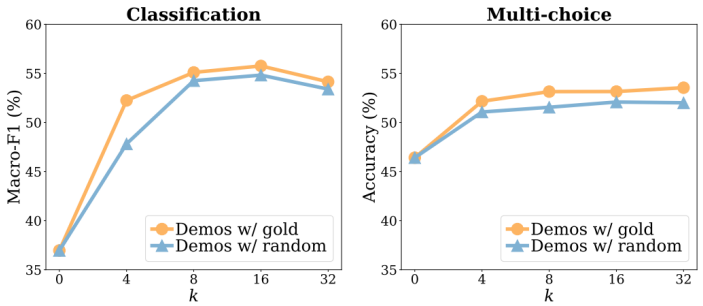


Figure 5: Ablations on varying numbers of examples in the demonstrations (k). Models that are the best under 13B in each task category (Channel MetaICL and Direct GPT-J, respectively) are used.

Source: http://arxiv.org/abs/2202.12837

Few-shot In-context Learning with LLMs

- Min et al. (2022) identified four meaningful factors:

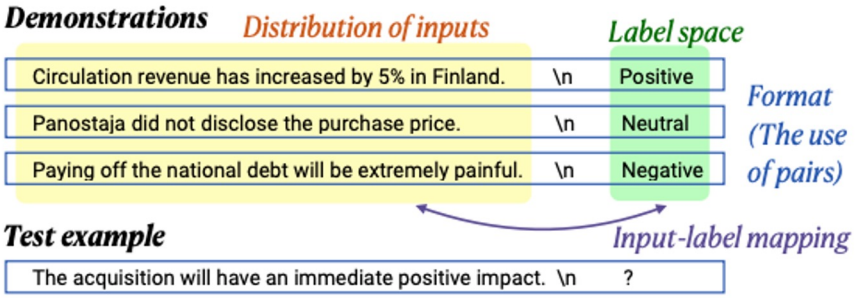


Figure 7: Four different aspects in the demonstrations: the input-label mapping, the distribution of the input text, the label space, and the use of input-label pairing as the format of the demonstrations.

- Another potentially meaningful aspect of in-context learning: what *exactly* are we asking the LLM?

Source: http://arxiv.org/abs/2202.12837

Prompt Engineering

- Not all prompts are equally good!
- Example: zero-shot news article classification using OPT-175B on the AG News dataset

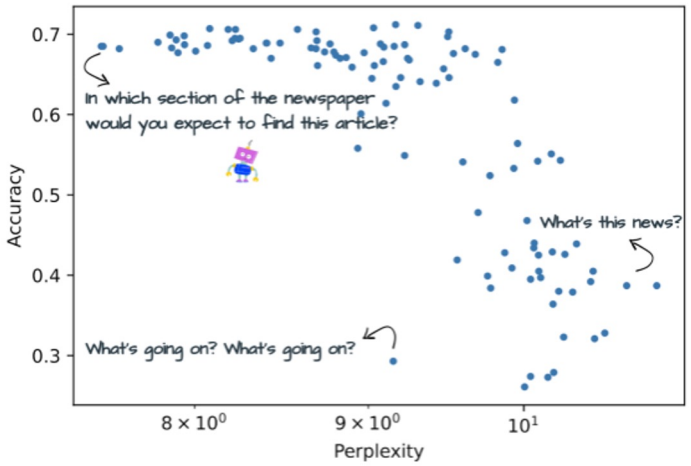
Prompt	Accuracy
What is this piece of news regarding?	40.9
What is this article about?	52.4
What is the best way to describe this article?	68.2
What is the most accurate label for this news article?	71.2

- What affects the accuracy associated with using a prompt?
- One potential answer: how likely the prompt is under the learned model's implied distribution over sequences

Source: https://arxiv.org/pdf/2212.04037

Prompt Engineering

- Not all prompts are equally good!
- Example: zero-shot news article classification using OPT-175B on the AG News dataset



- Perplexity is the exponentiated average negative log-likelihood of a sequence
- Lower perplexity = higher likelihood

Source: https://arxiv.org/pdf/2212.04037

Prompt Engineering

- Not all prompts are equally good!
- Example: zero-shot news article classification using OPT-175B on the AG News dataset

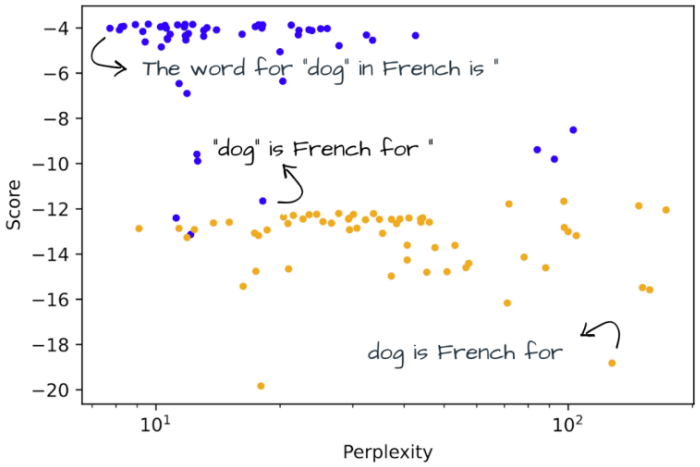


Figure 2: Score of correct label vs. perplexity for the word-level translation task in French with OPT 175B. The x axis is in log scale. The blue points stand for prompts with quotation marks for the words, while the yellow points are of prompts without quotation marks.

- Perplexity is the exponentiated average negative log-likelihood of a sequence
- Lower perplexity = higher likelihood

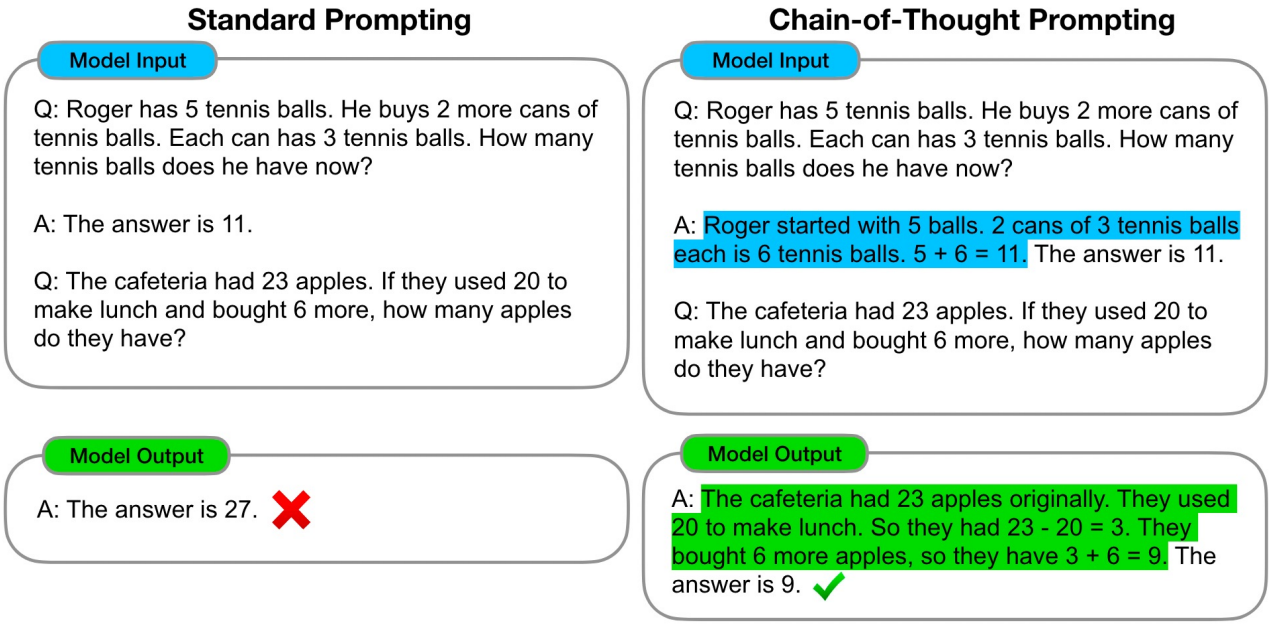
Source: https://arxiv.org/pdf/2212.04037

Learning to Prompt

- Some ways of *learning* better prompts for your task:
  1. Prompt paraphrasing – programmatically generate and test many different prompts from a paraphrase model, then pick the one that works best
  2. Gradient-based search – use optimization to search for the discrete representation of the prompt that makes the desired output most likely
  3. Prompt tuning – directly optimize the embeddings that are input into the LLM, without bothering to construct a discrete representation of the prompt

Chain-of-Thought Prompting

- Insight: asking an LLM to reason about its answer can improve its in-context performance
- Chain-of-thought prompting provides examples of reasoning in the in-context training examples



Source: http://arxiv.org/abs/2201.11903

Chain-of-Thought Prompting

- Insight: asking an LLM to reason about its answer can improve its in-context performance
- Chain-of-thought prompting provides examples of reasoning in the in-context training examples

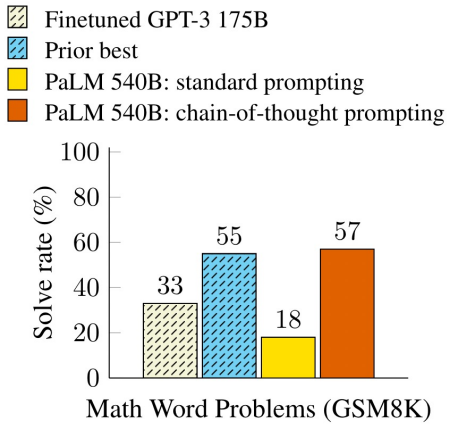


Figure 2: PaLM 540B uses chain-of-thought prompting to achieve new state-of-the-art performance on the GSM8K benchmark of math word problems. Finetuned GPT-3 and prior best are from Cobbe et al. (2021).

Chain-of-Thought Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

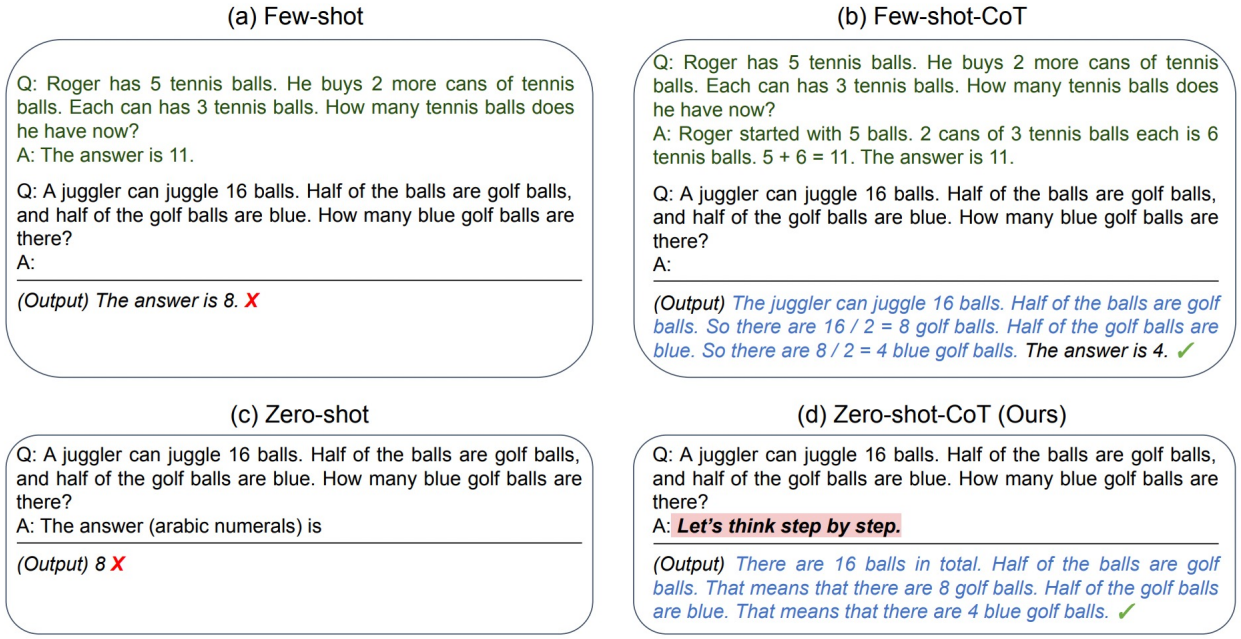
Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had 23 - 20 = 3. They bought 6 more apples, so they have 3 + 6 = 9. The answer is 9. ✓

Source: http://arxiv.org/abs/2201.11903

Chain-of-Thought Prompting

- Insight: literally just asking an LLM to reason about its answer can improve its in-context performance
- Chain-of-thought prompting provides examples of reasoning in the in-context training examples



Source: https://arxiv.org/pdf/2205.11916.pdf

Chain-of-Thought Prompting

- Insight: literally just asking an LLM to reason about its answer can improve its in-context performance
- Chain-of-thought prompting provides examples of reasoning in the in-context training examples

	MultiArith	GSM8K
Zero-Shot	**17.7**	**10.4**
Few-Shot (2 samples)	33.7	15.6
Few-Shot (8 samples)	33.8	15.6
Zero-Shot-CoT	**78.7**	**40.7**
Few-Shot-CoT (2 samples)	84.8	41.3

Source: https://arxiv.org/pdf/2205.11916.pdf

Chain-of-Thought Prompting

- Insight: asking an LLM to reason about its answer can improve its in-context performance
- Chain-of-thought prompting provides examples of reasoning in the in-context training examples

Source: https://openai.com/index/learning-to-reason-with-llms/