Solutions

10-423/10-623 Gen AI	**Name:**
Fall 2025	**Andrew ID:**
Practice Questions	**Room:**
11/10/25	**Seat:**
Time Limit: NA	**Exam Number:**

Instructions:

- Verify your name and Andrew ID above.
- This exam contains 43 pages (including this cover page).
  The total number of points is 167.
- Clearly mark your answers in the allocated space. If you have made a mistake, cross out the invalid parts of your solution, and circle the ones which should be graded.
- Look over the exam first to make sure that none of the 43 pages are missing.
- No electronic devices may be used during the exam.
- Please write all answers in pen or *darkly* in pencil.
- You have NA to complete the exam. Good luck!

1 AutoDiff / RNN-LMs (11 points)

1.1. (1 point) **Select one:** What is the primary purpose of using the chain rule of probability in language models?

- ◯ To convert raw text into a sequence of tokens
- ◯ To generate new tokens from a fixed-length vector
- ◯ To improve the accuracy of word embeddings
- ◯ To decompose the joint probability of a sequence of words into a product of conditional probabilities
- ◯ To calculate the probability of each token independently of the others

To decompose the joint probability of a sequence of words into a product of conditional probabilities

1.2. (1 point) **Select one:** Which of the following best describes the main purpose of backpropagation in neural networks?

- ◯ Backpropagation is used to compute the output of the neural network during inference.
- ◯ Backpropagation is used to update the weights of the network by propagating errors from the output layer to the input layer using the chain rule of calculus.
- ◯ Backpropagation is a technique to regularize the model and prevent overfitting.
- ◯ Backpropagation ensures that the neural network outputs discrete labels instead of continuous values.

B Backpropagation is used to update the weights of the network by propagating errors from the output layer to the input layer using the chain rule of calculus

1.3. (2 points) **Select multiple:** Which of the following are **true** about Recurrent Neural Networks (RNNs) in language modeling?

- ☐ RNNs can process sequences of variable length.
- ☐ RNNs maintain context through hidden states.
- ☐ RNNs do not suffer from vanishing and exploding gradient problems.
- ☐ RNNs are capable of capturing long-term dependencies in data, but may struggle with them without architecture modifications.
- ☐ RNNs use fixed-size input windows to handle sequential data.

A. RNNs can process sequences of variable length
B. RNNs maintain context through hidden states
D. RNNs are capable of capturing long-term dependencies in data, but may struggle with them without modifications.

1.4. (3 points) **Select all that apply:**

- ◯ One motivation of LSTMs is to improve the vanishing gradient problem seen with RNNs.
- ◯ It is not possible for LSTMs to experience the vanishing gradient problem.
- ◯ It is possible for LSTMs to experience the vanishing gradient problem.
- ◯ It is not possible for LSTMs to experience the exploding gradient problem.
- ◯ It is possible for LSTMs to experience the exploding gradient problem.

A, C, E

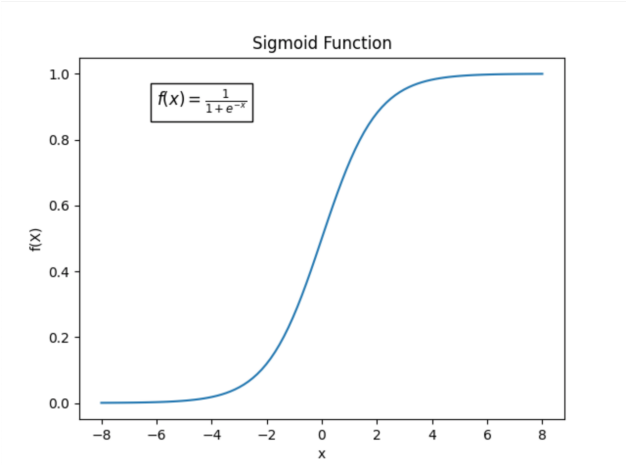
1.5. (1 point) **Select all that apply:** Vanishing gradients can cause weights to shrink uncontrollably, leading to numerical instability. Which of the following techniques is commonly used to mitigate vanishing gradients?

- ☐ activation functions such as ReLU
- ☐ batch normalization
- ☐ layer normalization
- ☐ gradient clipping
- ☐ for an RNN: use LSTM or GRU as the recurrent units instead of simple RNNs

A, B, D, E

In general, layer norm can be a cause of vanishing gradients (https://arxiv.org/pdf/2206.00330) and batch norm can help to address it.

1.6. Consider the sigmoid activation function:



(a) (1 point) **Short answer:** What would the gradient of the sigmoid be for a very large input?

close to zero

(b) (1 point) **Short answer:** Is this a potential problem when training an RNN? Explain.

vanishing gradients problem

(c) (1 point) **Short answer:** How does ReLU activation (ReLU(z) = max(0,z)) compare to sigmoid activation? Does it solve the problem above completely(if any)?

ReLU is piecewise linear
does not solve completely, but can help

2 Transformers and LLMs (30 points)

2.1. (1 point) **Select one:** Which of the following statements about Transformer Language Models is true?

- ◯ Transformer computation graphs grow linearly with the number of input tokens
- ◯ Residual connections in Transformers help in solving the vanishing gradient problem
- ◯ Layer normalization in Transformers helps mitigate internal covariate shift during training
- ◯ Transformers use a single attention head to process input sequences

B: Residual connections in Transformers help in solving the vanishing gradient problem
C: Layer normalization in Transformers helps mitigate internal covariate shift during training

2.2. (2 points) **Select one:** What is the primary reason for using multi-head attention instead of single-head attention in Transformer models?

- ◯ To allow the model to compute attention more efficiently.
- ◯ To capture different aspects or patterns of the input sequences by using multiple attention heads.
- ◯ To reduce the overall complexity of the Transformer model.
- ◯ To enable the model to focus solely on the most important tokens in the sequence.

B. Each head performs its own attention computation allowing the model to focus on different parts of the input sequence simultaneously and capture diverse features and relationships within the data.

2.3. (2 points) **Select one:** What is the main reason that adding residual connections to a transformer is useful?

- ◯ Improve the long-term memory of the model by helping to remember information from past timestamps.
- ◯ Instead of having to learn a complex transformation, they allow the model to learn an **additive** modification of the input.
- ◯ Instead of having to learn a complex transformation, they allow the model to learn an **multiplicative** modification of the input.

◯ To fix the issue that attention is position invariant by learning embeddings related to the position of each word in the input.

B This is the main motivation of residual connections

2.4. (2 points) **Select Multiple:** Which of the following is true regarding Transformer models (as in the "Attention is all you need paper")?

☐ After concatenating the outputs of all the heads we can directly send them to the Add & Norm step.

☐ The first token of the decoder model is the last hidden state of the encoder.

☐ The positional encodings have the same dimension as the token embeddings.

☐ The final weight matrix used to obtain the logits from the final hidden value is a new linear layer in the model.

C This is the only true option.

A - We multiply by outword project matrix W_O to combine the heads before going to the next step.
B - the first token for the decoder is a `<BOS>` kind of token, the hidden values of the encoder are only used in cross attention.
D - No we use weight-tying to use the same embedding matrix we used to get the embeddings from the token.

2.5. For the questions below, write the correct option from the list of models below -

1. An encoder model
2. A decoder model
3. A sequence to sequence model

(a) (1 point) Which of the above models would you use to complete prompts with generated text?

2

(b) (1 point) Which of the above models would you use for summarizing text?

3

(c) (1 point) Which of the above models would you use for classifying text inputs according to certain labels?

1

2.6. (1 point) **Select the correct answer:** What is one of the main computational challenges in scaling Transformers for large language models?

- ◯ The fixed size of the positional encoding limits the model's ability to process long sequences.
- ◯ The quadratic complexity of the self-attention mechanism with respect to sequence length.
- ◯ Overfitting due to the large number of parameters.
- ◯ The inability to parallelize model training.

B

2.7. (1 point) **Select the correct answer:** In self-attention, each token of the sequence:

- ◯ pays attention to itself
- ◯ pays attention to all words in the sequence
- ◯ pays attention all other tokens in the sequence except itself
- ◯ pays attention to a few tokens in its neighbourhood

B

2.8. (2 points) **Select all that apply:** Which of the following statements is true about LLM training?

- ☐ Since LSTMs generate one token at a time, their training across time steps cannot be parallelized
- ☐ Since transformers generate one token at a time, their training across time steps cannot be parallelized.
- ☐ If a sentence is padded with pad tokens, the transformer learns to generate the pad tokens.
- ☐ None of the above

A

2.9. (1 point) **True or False:** Computing attention values in a local neighbourhood over the input is the same as applying a convolution filter over the input

- ◯ True
- ◯ False

False. a convolution filter always applies the same transformation over all neighbours of every input, whereas attention scores of a local attention matrix will change values based on the input and its neighbours.

2.10. (5 points) **One of your friends is trying to understand how Rotary Positional Embeddings (RoPE) works.** Your friend makes the following statements but is not sure if they are correct.

Identify whether each statement is true or false, and explain why. If the statement is false, revise it to reflect the correct understanding of RoPE.

(a) (1 point) RoPE encodes positional information using sinusoidal functions similar to absolute positional embeddings but instead of adding them to the token embeddings, it applies them through rotation.

False. RoPE does rely on sinusoidal functions to encode positional information, but does NOT rotate positional encodings onto token embeddings. Instead, RoPE applies the rotations on the query and key vectors.

(b) (1 point) RoPE requires no additional learnable parameters to work.

True. RoPE uses predefined rotaional angles, meaning no extra parameters are trained.

(c) (1 point) The rotation matrix used in RoPE depends on m and i, where m is the token position and i is the index of the corresponding attention head.

False. Depends on m and i, but i is index within the embedding dimension, not attention head index.

(d) (1 point) For RoPE to work on arbitrary sequence lengths, it must be trained on sequences of all possible lengths during pretraining.

False. RoPE generalizes without training on all sequence lengths because its positional encoding is computed deterministically using a fixed formula.

(e) (1 point) RoPE relies on rotation matrices that remain the same across different layers of a transformer model.

True. For example, if d=2, across all transformer layers, the following rotation matrix will be applied:

$$R_{\theta,m} = \begin{bmatrix} \cos(m\theta) & -\sin(m\theta) \\ \sin(m\theta) & \cos(m\theta) \end{bmatrix}$$

2.11. **Sliding Window Attention** is an efficient variant of self-attention where each token only attends to a **fixed-size local window** of nearby tokens.

Consider a sequence of **length** $N = 5$ **with a sliding window size** $w = 3$, where w is the number of tokens in the window. The attention mechanism applies a mask such that each token can only attend to itself and other tokens within the window.

Answer the following questions:

(a) (2 points) **Short answer:** In a naive implementation, we apply a mask M to the calculated attention scores, such that the output $X' = \text{softmax}(\frac{QK^T}{\sqrt{d_k}} + M)V$. Construct the attention mask M, assuming:

1. A **non-causal** sliding window, where each token can attend to other token to its left and right within the window.

$$(M_1 =) \begin{bmatrix} 0 & 0 & -\infty & -\infty & -\infty \\ 0 & 0 & 0 & -\infty & -\infty \\ -\infty & 0 & 0 & 0 & -\infty \\ -\infty & -\infty & 0 & 0 & 0 \\ -\infty & -\infty & -\infty & 0 & 0 \end{bmatrix}$$

2. A **causal** sliding window, where each token can only attend to past tokens (including itself) within the window.

$$(M_2 =) \begin{bmatrix} 0 & -\infty & -\infty & -\infty & -\infty \\ 0 & 0 & -\infty & -\infty & -\infty \\ -\infty & 0 & 0 & -\infty & -\infty \\ -\infty & -\infty & 0 & 0 & -\infty \\ -\infty & -\infty & -\infty & 0 & 0 \end{bmatrix}$$

(b) (1 point) **Short answer:** Using the most efficient implementation, what is the time complexity of sliding window attention (in terms of **both** w and N)?

$O(N.w)$

(c) (2 points) **Short answer:** Briefly discuss one major advantage and one drawback of using sliding window attention instead of full-self attention?

1. **Advantage:** *Lower computational cost and memory usage*, allowing models to process longer sequences efficiently.
2. **Drawback:** *Limited receptive field*, meaning long-range dependencies are harder to capture, which may degrade performance for tasks requiring global context.

2.12. **Attention** Below are three expressions from the Homework 1 writeup that describe Grouped Query Attention (GQA). As a reminder, h_{kv} is the number of KV-heads, h_q is the number of query heads, and g is the group size.

$$\mathbf{S}^{(i,j)} = \mathbf{Q}^{(i,j)}(\mathbf{K}^{(i)})^T/\sqrt{d_k}, \quad \forall i \in \{1,\ldots,h_{kv}\}, \forall j \in \{1,\ldots,g\}$$
$$\mathbf{A}^{(i,j)} = \text{softmax}(\mathbf{S}^{(i,j)}), \quad \forall i \in \{1,\ldots,h_{kv}\}, \forall j \in \{1,\ldots,g\}$$
$$\mathbf{X}'^{(i,j)} = \mathbf{A}^{(i,j)}\mathbf{V}^{(i)}, \quad \forall i \in \{1,\ldots,h_{kv}\}, \forall j \in \{1,\ldots,g\}$$

Modify and rewrite these three expressions such that they describe the calculations for **multi-query attention**. Do not include g in your final answer.

$$\mathbf{S}^{(i)} = \mathbf{Q}^{(i)}(\mathbf{K})^T/\sqrt{d_k}, \quad \forall i \in \{1,\ldots,h_q\} \tag{21}$$
$$\mathbf{A} = \text{softmax}(\mathbf{S}^{(i)}), \quad \forall i \in \{1,\ldots,h_q\} \tag{22}$$
$$\mathbf{X}' = \mathbf{AV} \tag{23}$$
$$\tag{24}$$

3 Learning neural language models / Decoding (8 points)

3.1. (1 point) **Select one:** Which of the following statements is **incorrect**?

- ◯ RNNs process tokens sequentially, while Transformers process tokens in parallel.
- ◯ RNNs suffer from vanishing gradients for long dependencies, while Transformers can capture long-range dependencies using attention mechanisms.
- ◯ RNNs have difficulty with parallelization, whereas Transformers are highly parallelizable, allowing faster training.
- ◯ RNN language models inherently capture the sequential order of information, while transformer models do not have this built-in capability.

D. Transformers use positional encoding or embeddings to explicitly encode the positions of tokens in the sequence.

3.2. (2 points) **Select multiple:** Which of the following statements is/are **correct**?

- ☐ Neural language models are typically trained using a maximum likelihood estimation (MLE) approach.
- ☐ Greedy decoding selects the token with the highest probability at each step, and always returns the sequence with highest probability.
- ☐ Beam search decoding explores multiple possible sequences at each step and retains the top beam size sequences based on their cumulative probabilities (product of probabilities).
- ☐ During training, language models are optimized directly for decoding accuracy using beam search.

A. Neural language models are typically trained using a maximum likelihood estimation (MLE) approach.
C. Beam search decoding explores multiple possible sequences at each step and retains the top beam size sequences based on their cumulative probabilities (product of probabilities).

3.3. (3 points) **Select all that are true:**

- ☐ Storing previously computed keys and values across timesteps can help avoid redundant calculations, potentially improving efficiency.
- ☐ Because attention computation for each timestep is dependent on previous timesteps, scaled dot-product attention cannot easily be parallelized.
- ☐ Subword tokenization can help with computational tractability in comparison with other tokentization methods, while eliminating OOV words.

- ☐ As a transformer LM computes attention based on previous tokens, its computation graph grow linearly as the number of input tokens grows.

A. Key-value caching: keys and values are re-used over many timesteps so we can cache them for faster access.
C. Character-based tokenization suffers from computational issues due to longer sequences, and word-based tokenization leads to many OOV words, which subword tokenization solves.

3.4. (2 points) **Select all that are true:**

- ☐ A higher perplexity indicates we have a better language model.
- ☐ Greedy decoding guarantees to get the highest probability sequence.
- ☐ Given a LM we can find the probability of any sentence under it.
- ☐ The internal parameters of a Transformer LM are not dependent on the sequence length used during training (excluding PE and input embedding steps).

C, D
For D - None of the parameters in QKV, MLP, LayerNorm etc depend on sequence length. As long as you handle the Positional Embeddings in the input. Note accuracy will suffer greatly though if you go beyond the training seq length.

A - Lower is better
B - That is not the global optimal

4 Pre-training, fine-tuning / Modern Transformers (17 points)

4.1. (1 point) **Select one:** Which of the following is **NOT** one of the motivations for LoRA.

- ☐ Large language models are intrinsically low-dimensional.
- ☐ Adapters add latency at test time.
- ☐ Prefix tuning results in improvements to the model that do not monotonically increase with the number of parameters.
- ☐ None of the above

None of the above

4.2. (1 point) **Select one:** Which of the following best describes the rank in LoRA.

- ☐ The number of layers in the model.
- ☐ The dimension of the low-rank matrices A and B.
- ☐ The batch size during training
- ☐ The learning rate during finetuning

B

4.3. (1 point) **Select all that apply:** Compared to full fine-tuning, LoRA typically:

- ☐ Increases the total number of trained parameters.
- ☐ Increases the computation time during training.
- ☐ Decreases the total number of trained parameters.
- ☐ Requires more data to train.
- ☐ Decreases the GPU memory required during training.
- ☐ Improves interpretability of the final model.

C, E

4.4. (1 point) In the context of parameter efficient fine-tuning, how do adapter modules achieve parameter efficiency while still allowing for efficient fine-tuning?

Bottleneck dimension $m \ll d$

4.5. (2 points) **Select all that are true:** Which of the following are true regarding Prefix-Tuning.

- ☐ Prefix-tuning involves pre-pending task-specific vectors to the input.
- ☐ Prefix tuning is typically used to train a single set of parameters that are shared across multiple tasks.
- ☐ Prefix-tuning requires storing a separate tuned copy of the model for each task, leading to increased storage overhead as the number of tasks increases.
- ☐ None of the above

None of the above

4.6. (2 points) **Select one**: What is the primary purpose of Rotary Position Embeddings (RoPE) in a transformer model?

- ◯ To improve the efficiency of inference for a transformer model.
- ◯ The encode positional information in a way that is rotation invariant.
- ◯ To reduce the dimensionality of the input data.
- ◯ To improve the model's ability to retain long-range dependencies.

B

4.7. (2 points) **Select all that are true:** What is the primary advantage of using convolutional layers in neural networks, especially for image-related tasks?

- ◯ Convolutional layers reduce the number of parameters by sharing weights across the input.
- ◯ Convolutional layers ensure the model is invariant to transformations like rotation and scaling.
- ◯ Convolutional layers allow the model to learn feature hierarchies by focusing on local spatial regions.
- ◯ Convolutional layers remove noise from the input data during training.

A. Convolutional layers reduce the number of parameters by sharing weights across the input.
C. Convolutional layers allow the model to learn feature hierarchies by focusing on local spatial regions.

4.8. (3 points) **Select one:** Which of the following is not true about auto-encoders?

- ◯ Encoders use a non linear activation function to compute an encoding of the input, which the decoder uses to compute a reconstruction of the input.
- ◯ The output of an autoencoder is exactly the same as its input.
- ◯ Autoencoders are an unsupervised learning technique.
- ◯ Autoencoders can be trained by using the same backpropagation algorithm used for a 2-layer neural network, where the loss is propagated back from the reconstruction error to update both the encoder and decoder weights.

B. The input and output of an autoencoder is similar, but not exactly the same, which is what the loss seeks to model.

4.9. (1 point) Say a transformer model takes 1 day to train with a dataset of sequence length N, how long would it take to train the same model if the sequence length were increased to $2N$ assuming everything else remained exactly the same?

4 days because the time complexity of attention is of the order N^2. In other words the new computation would be of the order $(2N)^2 = 4N^2$ which is 4 times the initial training time.

4.10. (1 point) **Select one:** Which of the following best describes the primary advantage of using Flash Attention in transformer models?

- ◯ It reduces the number of layers required for effective training.
- ◯ It accelerates attention computation by optimizing memory reads and writes.
- ◯ It replaces the softmax operation in the attention mechanism with a more efficient function.
- ◯ It improves model stability by enhancing attention score calculation.

B. It accelerates attention computation by optimizing memory reads and writes.

4.11. (2 points) **Select all that are true:** Which of the following are true regarding Prefix Tuning ?

- ☐ With a pre-trained LLM, we can attach and swap task-specific prefix parameters at inference time for different tasks.
- ☐ The dimension of the learned prefix is not the same as the input embeddings. It requires a linear transformation.

- ☐ We are essentially letting the model learn what is the correct prompt for this task.
- ☐ Prefix Tuning is just a new way to create vector embeddings from input tokens.

A,C are the correct options

B - It's the same
D - No it doesn't replace how input token embeddings are created.

5 Vision Transformers (14 points)

5.1. (2 points) **Select all that are true** In a convolutional neural network:

- ☐ Backpropagation cannot be applied over a max-pool layer as the max function is not differentiable
- ☐ Backpropagation cannot be applied over a downsampling layer as the downsampling function is not differentiable
- ☐ A 4x4 convolutional kernel over RGB inputs has 16 weights
- ☐ A 4x4 convolutional kernel over RGB inputs has 48 weights

D

5.2. (2 points) **Select One:** We have the following image and want to apply the following 3×3 kernel, with a padding of 1 and stride of 2. After applying this kernel, what is the value at position $(0, 1)$ (0-indexed)?

$$\begin{bmatrix} 1 & 2 & 2 & 1 & 1 \\ 2 & 4 & 4 & 2 & 3 \\ 1 & 3 & 3 & 1 & 4 \\ 1 & 2 & 3 & 1 & 3 \\ 0 & 1 & 4 & 2 & 0 \end{bmatrix}$$

Figure 1: Image Matrix

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Figure 2: Kernel Matrix

- ◯ 1
- ◯ 2
- ◯ -1
- ◯ 0

-1

$$\begin{bmatrix} 0 & 0 & 0 \\ 2 & 2 & 1 \\ 4 & 4 & 2 \end{bmatrix} * \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} = 2 + 2(-4) + 1 + 4 = -1$$

5.3. (1 point) In a Vision Transformer (ViT) model, which is an encoder-only transformer network, the extra learnable [class] embedding is analogous to the [CLS] token in a BERT model. It is used to aggregate the information from the entire image into a single vector, which can be used for many downstream tasks such as object detection, etc.
True or False: The use of the extra learnable [class] embedding at the beginning of the input sequence, in principle, will lead to a better-performing model than if the extra learnable [class] embedding was used at the very end of the input sequence.

- ◯ True
- ◯ False

False

5.4. Consider an application of attention to an image.

(a) (2 points) **Select all that apply:** For a flattened sequence $X \in \mathbb{R}^{N \times C}$ of $N = H \times W$ pixels of an image with height H, width W, and C-dimensional features, the attention scores are given by $A = \text{softmax}(\frac{QK^T}{\sqrt{D}})$, where Q, K are D-dimensional linear projections of the features: $Q = XW_q$, $K = XW_k$, with $W_q, W_k \in \mathbb{R}^{C \times D}$. If we do not scale the attention scores by $\sqrt{D}$:

- ☐ We might get instability issues during training.
- ☐ The attention scores will not be normalized.
- ☐ We will not use information from nearby pixels.
- ☐ The dot product will have a larger variance.

A, D

(b) (2 points) **Select the correct answer:** We now subsample in the spatial dimensions the image of the previous question by a max pooling layer with a stride of 2 and taking the maximum over a window of size 2×2, to obtain the flattened sequence $\hat{X}$. We redefine the keys such as $K = \hat{X}W_k$. The self-attention can then be written as $S = A \times V$, where $V = \hat{X}W_v$ with $W_v \in \mathbb{R}^{D \times D'}$. If we use the subsampled version $\hat{X}$ when defining the keys and values, we reduce the computational cost of the self-attention by a factor of:

- ◯ 1/2
- ◯ 1/4
- ◯ 1/16
- ◯ 1

The flattened sequence is now 1/4 of the size. Since attention is quadratic in the length of the sequence we reduce the cost by 1/16.

5.5. (2 points) **Select all that apply:** Compared to Convolutional Neural Networks (CNNs), Vision Transformers (ViT):

- ☐ Rely exclusively on convolutional operations for feature extraction.
- ☐ Can capture global context in images more effectively through attention mechanisms.
- ☐ Typically require larger datasets for effective training.
- ☐ Maintain constant computational complexity irrespective of image resolution.

B, C

5.6. (2 points) **Select all that apply:** Consider an image processed by a Vision Transformer. If you remove the classification token ([CLS]) embedding:

- ☐ The model becomes incapable of performing any form of classification.
- ☐ You could still classify images using alternative pooling strategies.
- ☐ Self-attention between image patches remains unaffected.
- ☐ Positional embeddings would become irrelevant.

B

5.7. (1 point) **Explain briefly:** Suppose you modify a Vision Transformer to apply attention only to a limited local neighborhood around each patch instead of globally across all patches. What impact might this have on the model's performance?

Limiting attention to local neighborhoods might reduce computational complexity and memory requirements, making the model more efficient. However, it could decrease the model's ability to capture long-range dependencies and global contextual information, potentially lowering performance on tasks requiring broader context understanding.

6 Generative Adversarial Networks (GANs) (9 points)

6.1. (1 point) **True or False:** In the training process of a Generative Adversarial Network (GAN), if the input noise vector z to the generator G_θ is completely random and lacks any discernible pattern, it becomes impossible to perform backpropagation through the generator G_θ.

- ◯ True
- ◯ False

False

6.2. (1 point) **Select One:** Which of the following statements best describes the role of the discriminator in a Generative Adversarial Network (GAN)?

- ◯ The discriminator generates new data samples from the latent space.
- ◯ The discriminator updates the generator's parameters to improve the quality of generated samples.
- ◯ The discriminator distinguishes between real and generated data samples, providing feedback to the generator.
- ◯ The discriminator minimizes the difference between real and generated data distributions.

C

6.3. (2 points) **Select all that apply:** What could serve as input to the generator in a GAN?

- ☐ noise
- ☐ noise and label
- ☐ text description
- ☐ image

A, B, C, D

6.4. (1 point) **True or False:** If the discriminator overfits on the training data, the generator will produce samples that closely resemble the training images.

- ◯ True
- ◯ False

True

6.5. (2 points) **Explain briefly:** Suppose you modified the GAN by feeding real images occasionally into the generator instead of random noise. How might this impact the training process and the quality of generated images?

Feeding real images occasionally into the generator instead of random noise could cause the generator to rely on copying existing features rather than generalizing new image generation. This might initially improve visual quality but would significantly reduce diversity and hinder the model's ability to generalize and capture complex distributions of data.

6.6. (2 points) **Select all that apply:** Consider a class-conditional GAN trained on highly imbalanced classes. Which of the following challenges might arise?

- ☐ Increased computational cost during inference
- ☐ Higher likelihood of discriminator overfitting on majority classes
- ☐ Difficulty in generating realistic minority class samples
- ☐ Higher probability of mode collapse in minority classes

B, C, D

7 Variational Autoencoders (VAEs) (8 points)

7.1. (0 points) **Select all that apply:** Which of the following are true statements about KL divergence?

- ☐ KL divergence is symmetric: $\mathrm{KL}(p||q) = \mathrm{KL}(q||p)$.
- ☐ $\mathrm{KL}(p||q)$ is minimized when distributions p, q are equivalent, i.e. $q(x) = p(x)$ for all $x \in X$.
- ☐ Maximizing ELBO is the same as minimizing KL divergence.
- ☐ $0 \leq \mathrm{KL}(p||q) \leq 1$ for all probability distributions p and q.
- ☐ None of the above.

B, C

7.2. (1 point) **True or False:** In Variational Autoencoders (VAEs), we sample from the latent distribution to generate the output. Since sampling prevents backpropagation through the network, neural networks cannot be used to build VAEs.

- ◯ True
- ◯ False

False

7.3. (2 points) **Select all that apply:** The following statements of KL-divergence are correct:

- ☐ KL-divergence measures the proximity of two distribution
- ☐ KL-divergence is maximized when two distribution are identical
- ☐ KL-divergence is not symmetric: $KL(q||p) \neq KL(p||q)$
- ☐ KL-divergence is always non-negative

A, C, D

7.4. (1 point) **Short answer:** Explain the intuitive meanings of two terms in the VAEs loss function: $L = \sum_i -\mathbb{E}_{q_\theta(z|x^{(i)})}[\log p_\phi(x^{(i)}|z)] + KL(q_\theta(z|x^{(i)})||p(z))$.

Reconstruction loss and Regularization.

7.5. (1 point) **Select all that apply:**Why must the noise scaling factor α_t in Diffusion Probabilistic Models (DDPMs) follow a schedule?

- ☐ To ensure smooth addition of noise, preventing excessive corruption of data early on.
- ☐ To allow the model to gradually learn how to denoise at different noise levels.
- ☐ To ensure the reverse process can recover the original data distribution effectively.
- ☐ To control the amount of noise added during each step, ensuring balanced corruption across all timesteps.
- ☐ None of the above.

A, B, C, D

7.6. (1 point) **Select One:** Suppose you are training a neural network $p(x)$ to approximate a known but hard-to-define probability distribution $q(x)$. We choose as our objective function $KL(q \parallel p) = \mathbb{E}_q\left[\log \frac{q(x)}{p(x)}\right]$. Which of the following describes how to select the optimal p(x) to minimize this objective function?

- ◯ Select p(x) to be as large as possible in order to minimize $\frac{q(x)}{p(x)}$.
- ◯ Select p(x) to be as small as possible in order to maximize $\frac{q(x)}{p(x)}$.
- ◯ Select p(x) to be as close as possible to q(x) in order to make $\frac{q(x)}{p(x)} = 1$.
- ◯ Select p(x) to be the uniform distribution so this term is independent of the parameters of the network and can be ignored.

C, log(1) = 0 and the KL divergence is non-negative. Therefore 0 is an optimal value.

7.7. (1 point) **Select One:** Let function f(x, N) be the function which takes an image x and adds Gaussian noise to the image N times such that the output is the Nth step of forward diffusion in a DDPM model. What is the time complexity of an optimal implementation of this function? Assume h and w are the image height and width.

- ◯ O(h*w)
- ◯ O(log(N)*h*w)
- ◯ O(N*h*w)

A. Adding Gaussian noise N times can be mathematically equivalently implemented by adding lots of Gaussian noise just once.

7.8. (1 point) Recall that the likelihood of the data can be expressed as

$$\log p(x) = KL(q_\phi(z) \parallel p(x,z)) + ELBO(q_\phi(z))$$

where q_ϕ is an encoder in a VAE.

Considering the above equation, why it is desirable for us to maximise ELBO when training a VAE?

Hint: What happens to the KL divergence term when we maximise ELBO?

A. Since $\log p(x)$ is constant with respect to ϕ, any maximisation of the ELBO would invoke an equal minimisation of the KL divergence term. Thus, it is guaranteed that the more we maximise the ELBO, the closer our approximate posterior will get to the true posterior.

8 Diffusion Models (8 points)

8.1. (1 point) **Select all that apply:** Which of the following will follow a Gaussian distribution?

- ☐ The sum of two Gaussians
- ☐ The difference of two Gaussians
- ☐ The conditional of a Gaussian with a Gaussian mean
- ☐ The conditional of a Guassian with a nonlinear mean
- ☐ None of the above

choice 1, choice 2, choice 3

8.2. (1 point) **Select all that apply:** Which of the following can NOT be used in the implementation of a diffusion model such as the DDPM or LDM?

- ☐ Rotational positional embeddings
- ☐ Sinusoidal positional embeddings
- ☐ A neural network
- ☐ A UNet model
- ☐ Self-attention
- ☐ A large language model (LLM)
- ☐ A variational autoencoder (VAE)
- ☐ None of the above

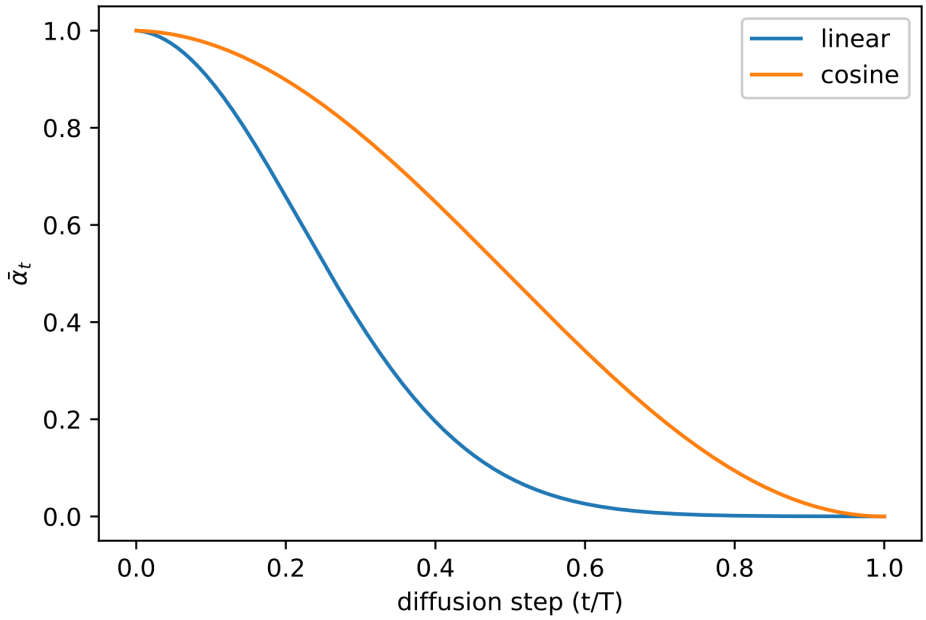
None of the above

8.3. (5 points) **Short Answer:** Consider the noise-based parameterization of a diffusion model: in this case, we use a neural network to estimate the noise (ϵ) added to the initial data point (x_0) to produce the current state (x_t). This noise-based parameterization has been empirically observed to yield images of higher quality compared to other parameterization methods.

How does this noise-based parameterization differ from the alternatives, and why might this formulation lead to the highest quality images?

Other parameterizations may estimate the moments of the noise (mean and variance) or approximate x_0 at each time step. These methods might not be as effective as they estimate the noise indirectly through by-products. In other words, the forward diffusion process only adds noise so the reverse diffusion process need only estimate the noise which was added.

8.4. (1 point) Recall from lecture the following graph that compares $\bar{\alpha}_t$ values when α_t follows a linear versus a cosine schedule.



Considering the given graph, why would using a cosine schedule rather than a linear schedule when training a diffusion model result in better performance?

Hint: How does the value of $\bar{\alpha}_t$ affect the proportion of the original image x_0 that remains at timestep t?

A. Low values of $\bar{\alpha}_t$ mean that very little information of the original image remains at timestep t. When this is the case, the model is unable to learn much meaningful information from the data since the images from one time step to the next are mostly just noise. The timesteps returned by a cosine schedule places more emphasis on higher values of $\bar{\alpha}_t$ compared to a linear schedule, thus allowing the model to learn more information at each timestep.

9 In-context Learning (8 points)

9.1. (2 points) **Select all that are true:**

- ☐ In-context learning can involve providing AI model with examples of the desired task within the input prompt.
- ☐ In-context learning requires fine-tuning the model on a specific dataset before it can perform the task.
- ☐ In-context learning allows the model to adapt to new tasks without the need for additional training.
- ☐ The performance of in-context learning depends on the model's ability to understand and generalize from the given examples.
- ☐ In-context learning is limited to tasks that the model has been explicitly trained on during its initial training phase.
- ☐ The choice and quality of the examples provided in the input prompt can significantly impact the model's performance in in-context learning.

A, C, D, F

9.2. (1 point) **True or False:** The performance of In-Context Learning is generally better in larger language models because they are trained on more diverse tasks.

- ◯ True
- ◯ False

True

9.3. (2 points) **True or False:** In-Context Learning can be thought of as a form of meta-learning, where the model implicitly learns to adapt to tasks through examples within the prompt.

- ◯ True
- ◯ False

True

9.4. (1 point) **Short answer:** Explain why In-Context Learning can struggle with tasks that require multi-step reasoning.

Multi-step tasks often exceed the model's context window, causing it to lose track of earlier steps or information necessary for completing later steps in the sequence.

9.5. (2 points) **What are key limitations of in-context learning? Select all that are true:**

- ☐ In-context learning is constrained by models' context window length
- ☐ In-context learning doesn't persist knowledge between sessions
- ☐ In-context learning adds more computational resources during inference
- ☐ In-context learning exhibits high sensitivity to prompt formulation
- ☐ In-context learning lacks reliable heuristics for determining the optimal number of examples needed to maximize model performance
- ☐ In-context learning is limited to tasks that were explicitly covered during the model's pre-training process.

A, B, C, D, E

10 RLHF / DPO (12 points)

10.1. (1 point) **True or False:** In the RLHF process, the reward model typically has more parameters than the language model being fine-tuned.

- ◯ True
- ◯ False

False

10.2. (1 point) **Select one:** Which of the following best describes the training process for the reward model in RLHF?

- ◯ It is trained to minimize the perplexity of human-written responses
- ◯ It is trained to maximize the likelihood of all possible responses
- ◯ It is trained so that higher-ranking responses receive larger rewards than lower-ranking ones
- ◯ It is trained to directly optimize the parameters of the language model

C

10.3. (1 point) **Select one:** During the RLHF process, what is the primary objective function being optimized when training the policy (language model)?

- ◯ To minimize the perplexity of human-labeled responses
- ◯ To maximize the expected reward from the reward model
- ◯ To minimize the KL divergence between model outputs
- ◯ To maximize the likelihood of all possible responses

B

10.4. (1 point) **Select all that apply:** What does the state space consist of in RLHF?

- ☐ The vocabulary of possible next tokens
- ☐ The set of all possible prompts
- ☐ All possible sequences of tokens
- ☐ The parameters of the language model

C

10.5. (2 points) **Select the correct answer** Which of the following best describes the primary goal of chain-of-thought prompting?

- ◯ To improve the model's ability to generate coherent and engaging stories
- ◯ To enhance the model's performance on tasks that require multi-step reasoning and problem-solving
- ◯ To reduce the computational resources required for training large language models
- ◯ To increase the model's capacity to understand and respond to emotional cues in natural language

B

10.6. (2 points) **Select all that are true:** In the context of Reinforcement Learning with Human Feedback (RLHF), what steps are involved in the process?

- ☐ Training a reward model based on human rankings of model-generated responses.
- ☐ Using reinforcement learning to train the model with rewards as "ground truth" for desirable outputs.
- ☐ Collecting prompts and generating multiple responses for each to be ranked by humans.
- ☐ Applying unsupervised learning techniques to automatically generate training data without human intervention.

A, B, C

10.7. (2 points) **Select all that are true:** Which of the following statements accurately describe aspects of Instruction Fine-Tuning?

- ☐ It aims to reduce the perplexity of a large training corpus by predicting the most likely next words in a sequence.
- ☐ It involves fine-tuning Language Models (LMs) on a dataset specifically created to align the model's output with human expectations for given tasks.
- ☐ It uses sources of prompts and responses to build a "chat agent" training dataset.
- ☐ It incorporates multiple names such as chat fine-tuning, alignment, and behavioral fine-tuning.

B, C, D

10.8. (2 points) **Short answer:** Explain how RLHF and DPO differ in their use of human-ranked data?

RLHF uses human-ranked data to train a reward model that is then used to train a language model with reinforcement learning. DPO instead uses human-ranked data to directly optimize LLM parameters by maximizing the likelihood of the human-ranked data.

11 Text-to-image Models and VLMs (19 points)

11.1. (2 points) **Select all that apply:** In Homework 4 (and very briefly in lecture) we saw an example of a text-to-image model that used "classifier free guidance". In this technique, motivated by the desire to add a temperature parameter to diffusion models, two diffusion models $\epsilon_\theta(\mathbf{x}_t, t, \tau_\theta(\mathbf{y}))$ and $\epsilon_\theta(\mathbf{x}_t, t)$ are jointly trained, where y represents a text string to condition upon, t a timestep, and x_t a partially noised image. "Classifier free guidance" adds a parameter w which interpolates between each models output at inference time such that $\tilde{\epsilon}_\theta(\mathbf{x}_t, t, \tau_\theta(\mathbf{y})) = (1-w)\epsilon_\theta(\mathbf{x}_t, t, \tau_\theta(\mathbf{y})) + (w)\epsilon_\theta(\mathbf{x}_t, t)$.

Which of the following would be the effect of generating a single image using this $\tilde{\epsilon}_\theta(\mathbf{x}_t, t, \tau_\theta(\mathbf{y}))$ (as compared to generating images using $\epsilon_\theta(\mathbf{x}_t, t, \tau_\theta(\mathbf{y}))$? Assume w is positive and less than 1.

- ☐ Less diversity in generated images
- ☐ Higher quality generated images
- ☐ Inference time doubled
- ☐ Worse text-image alignment

A, B, D

References: https://arxiv.org/pdf/2207.12598 https://arxiv.org/pdf/2205.11487

11.2. (2 points) **Select all that apply:** A Text-To-Image Latent Diffusion Model (LDM) pipeline includes the following components.

- ☐ image autoencoder
- ☐ text encoder
- ☐ UNet
- ☐ tokenizer
- ☐ discriminator

image autoencoder, text encoder, UNet, tokenizer

11.3. (2 points) **Select all that apply:** The following statements of classifier-free guidance are correct:

- ☐ Classifier-free guidance can improve the quality of generated samples.
- ☐ Larger guidance scale always leads to higher quality of generated samples.
- ☐ Larger guidance scale increases the diversity of generated samples.
- ☐ Larger guidance scale decreases the diversity of generated samples.

A, D

11.4. (1 point) **True or False:** In the LDM framework, the latent representations from the VAE can be directly fed to the diffusion UNet.

- ◯ True
- ◯ False

False

11.5. (1 point) **True or False:** CLIP separately trains an image encoder and a text encoder and only jointly inference to predict the correct pairings of (text, image) at test time.

- ◯ True
- ◯ False

False

11.6. (2 points) **Select all that apply:** Regarding the batch size in the CLIP (Contrastive Language-Image Pre-training) objective:

- ☐ The memory consumption for CLIP's contrastive loss scales quadratically ($O(B^2)$) with batch size B due to the similarity matrix.
- ☐ In CLIP's contrastive loss, a batch of size B creates a B×B similarity matrix with positive pairs on the diagonal and all other combinations serving as negative pairs.
- ☐ For a batch size of 32, CLIP's contrastive loss computation would utilize 32 positive pairs and 992 negative pairs.
- ☐ Increasing batch size in CLIP primarily benefits computation speed rather than model performance.

A, B, C
Option A is correct because CLIP's memory consumption grows quadratically with batch size. Option B is correct as CLIP creates a B×B matrix where correct pairs lie on the diagonal and all other combinations serve as negative samples. Option C is correct - with 32 samples, there are 32 positive pairs and 32^2-32=992 negative pairs. Option D is incorrect because larger batch sizes primarily benefit training by providing more synthetic negatives, improving model performance, not speed.

11.7. (2 points) **Select all that apply:** Select the correct statement about SigLIP compared to CLIP:

- ☐ SigLIP uses the same softmax-based loss as CLIP but increases the batch size to improve vision-language alignment.
- ☐ SigLIP increases batch size to avoid communication overhead in distributed training.
- ☐ SigLIP replaces the softmax in CLIP's objective function with a sigmoid to make distributed training more efficient.
- ☐ SigLIP scales more effectively to larger batch sizes while maintaining strong performance even with smaller batches.

C, D

11.8. (2 points) **Select all that apply:** Which statements about VQ-VAE are correct?

- ☐ The straight-through estimator treats the gradient w.r.t. $z_q(x)$ as an estimate of the gradient w.r.t. $z_e(x)$ to enable backpropagation through the non-differentiable quantizer.
- ☐ Its training objective includes a reconstruction loss and two regularization terms: $\|\text{sg}[z_e(x)] - z_q(x)\|_2^2$ (commitment loss) and $\beta\|z_e(x) - \text{sg}[z_q(x)]\|_2^2$ (codebook loss).
- ☐ The $\arg\min$ quantization step is differentiable, so exact gradients flow from the decoder to the encoder through $z_q(x)$.
- ☐ Unlike CLIP-based VLMs, VLMs with VQ-VAE encoders discretize images into codebook tokens and generate both image and text.

A, D
Option B is incorrect because $\|\text{sg}[z_e(x)] - z_q(x)\|_2^2$ is the codebook loss, which updates $z_q(x)$ to move closer to the encoder output $z_e(x)$. In contrast, $\beta\|z_e(x) - \text{sg}[z_q(x)]\|_2^2$ is the commitment loss, which updates the encoder output to commit to its assigned codebook vector. Option C is incorrect because the $\arg\min$ quantization step is non-differentiable.

11.9. (1 point) **True or False:** In the Diffusion Transformer (DiT), both increasing the Transformer size and decreasing the patch size lead to improved (lower) FID performance.

- ◯ True
- ◯ False

True

11.10. (4 points) **Short answer:** We have covered several text-to-image and vision-language models in class, including Parti, DALL·E 2, and PaliGemma. Briefly describe the

key architectural components and the training process of each model.

Parti is a text-to-image Transformer model. In stage one, it trains a VQ-VAE tokenizer that maps images into discrete visual tokens and can reconstruct images from those tokens at inference time. In stage two, it trains a sequence-to-sequence autoregressive Transformer that generates image tokens from text tokens.

DALL·E 2 is a diffusion-based text-to-image model built on CLIP embeddings. It first trains a CLIP model and keeps its text encoder frozen. A diffusion model is then trained to map text embeddings to image embeddings, followed by a decoder diffusion model that generates high-fidelity images from image embeddings.

PaliGemma is a vision-language model combining a SigLIP image encoder and a Gemma-2B decoder-only language model. SigLIP converts images into token sequences, and a linear projection aligns them with Gemma's text token space. The combined image–text tokens are fed into the Transformer decoder, which is trained across multimodal tasks.

12 Prompt2Prompt (19 points)

12.1. **Fill in the blank:** *In Prompt-to-Prompt editing, controlling the cross-attention maps allows for* _____ *while maintaining the overall structure of the source image.*

- ◯ generating entirely new images
- ◯ local edits to specific objects or attributes
- ◯ adding random noise to the image
- ◯ removing all semantic information from the image

local edits to specific objects or attributes

12.2. **Fill in the blank:** *In the Prompt-to-Prompt method, changes to the image are primarily achieved by modifying* _____.

- ◯ the latent noise vector
- ◯ the cross-attention maps
- ◯ the input image resolution
- ◯ the number of diffusion steps

the cross-attention maps

12.3. **True or False:** Using Prompt-to-prompt to edit real images (such as those captured by a digital camera or cell phone) is impossible.

- ◯ True
- ◯ False

False, see Prompt-to-prompt white paper section 4 "Applications" heading "Real Image Editing": https://arxiv.org/pdf/2208.01626

12.4. (1 point) **Select the correct answer:** In the context of attention replacement editing, why might a mapping matrix be required to be sparse (mostly zeros)?

- ◯ To reduce computational complexity
- ◯ To emphasize the modifications between the original and edited prompts
- ◯ To prevent overfitting during training
- ◯ To maintain attention on unchanged tokens while updating the changed ones

D

12.5. (2 points) **Select the correct answer:** How would attention re-weighting affect the generation of an edited image in diffusion models?

- ◯ It adjusts the emphasis on specific features in the image
- ◯ It alters the style of the image without affecting the content
- ◯ It changes the random seed influencing the image generation
- ◯ It increases the resolution of the generated image

A

12.6. (3 points) **Select the correct answer:** The trade-off in Prompt-to-Prompt image editing between fidelity to the edited prompt and the source image is controlled by:

- ◯ Adjusting the image contrast settings
- ◯ The number of injection timestamps
- ◯ Varying the amount of Gaussian noise added
- ◯ The depth of the U-Net used

B

12.7. (4 points) **Select the correct answer:** The process of adjusting attention weights in image editing via text prompts is analogous to which of the following?

- ◯ Fine-tuning the hyperparameters of a neural network
- ◯ Adjusting the focus in a photographic image
- ◯ Changing the storyline in a movie script
- ◯ Resizing an image while maintaining aspect ratio

B

12.8. (2 points) **Select the correct answer:** The attention mechanism in Prompt-to-Prompt is modified by:

- ◯ Only modifying the keys and values
- ◯ Only modifying the keys and query embeddings
- ◯ Removing attention layers altogether
- ◯ Adding noise at random timesteps

A

12.9. (2 points) **Select the correct answer:** In the second diffusion process for generating the edited image in Prompt-to-Prompt, how does the model condition on the prompts?

- ◯ The model conditions on the original prompt y throughout the entire reverse diffusion process.
- ◯ The model conditions on the edited prompt $\hat{y}$ for all timesteps.
- ◯ The model conditions on y for the first τ timesteps and switches to $\hat{y}$ afterward.
- ◯ The model randomly alternates between y and $\hat{y}$ at each timestep to balance consistency and novelty.

B

12.10. (1 point) **True or False:** In the Prompt-to-Prompt process, the LDM model is involved in both training and inference to adjust attention weights for image editing.

- ◯ True
- ◯ False

False. Prompt-to-Prompt is only an inference process. The model does not require training during the image editing process; it only modifies how the samples are drawn from the pre-trained latent diffusion model by adjusting the attention weights from previous runs.

12.11. (4 points) **Short answer:** Given the Edit function:

$$\text{Edit}(A_t, A_t^*, t) := \begin{cases} A_t & \\ & \text{if } t < \tau \\ A_t^* & \text{otherwise} \end{cases}$$

Please answer the following:

1. What role does τ play in the Prompt-to-Prompt process?
2. How would the attention weights evolve if τ were set to the maximum timestep T?
3. What would happen if τ were set to 0?
4. Do we always need to fine-tune τ for optimal results?

1. **Role of τ**: τ controls the timestep at which the cross-attention weights switch from the original prompt's stored weights to the new prompt's computed weights during image editing. This hyperparameter determines how much structural coherence (from the original prompt) versus semantic change (from the edited prompt) is preserved.
2. $\tau = T$ (maximum timestep): The model would use the original prompt's attention weights for all timesteps, effectively ignoring the new prompt. (Full Cross-Attention injection)
    - Result: Minimal or no semantic changes in the edited image, preserving the original structure entirely.
3. $\tau = 0$: The model would immediately switch to the new prompt's attention weights starting from timestep T.
    - Result: Complete semantic alignment with the edited prompt but loss of structural coherence with the original image. (No Cross Attention Injection)
4. **Fine-tuning τ**: τ is a hyperparameter that must be tuned based on the editing task. Better editing results can be achieved with experimentation - giving the optimal balance between the new edited object/attribute and the orignal structure of the image

13 Scaling Laws (4 points)

13.1. **Select One:** *What is the relationship between compute budget and data filtering, as suggested by recent scaling laws?*

- ◯ More compute always requires more data filtering
- ◯ As compute increases, less data filtering is needed
- ◯ Data filtering is independent of compute budget
- ◯ More compute requires exponentially more data filtering

As compute increases, less data filtering is needed

13.2. **Select One:** *What was the key finding regarding the balance between model size and training data found by the Hoffman et. al. study?*

- ◯ Increasing model size is more important than increasing training data
- ◯ The optimal ratio is to increase parameters 8x for every 5x increase in training data
- ◯ Previous models were using too little data relative to their parameter count
- ◯ Convergence is not critical for good performance

Previous models were using too little data relative to their parameter count

13.3. (2 points) **Short answer:** Why is the MoE model more efficient?

Only a small portion of the experts are activated during the inference.

13.4. (2 points) **Short answer:** Why is the MoE model difficult to train?

Difficult to balance the training of different experts.

13.5. **Select One:** *According to scaling law studies, how does the test loss of a generative model typically behave as the number of parameters increases?*

- ◯ It decreases linearly
- ◯ It decreases following a power-law relationship
- ◯ It remains constant regardless of parameter count
- ◯ It increases exponentially with more parameters

It decreases following a power-law relationship

13.6. **Select One:** *Why do scaling laws imply diminishing returns when simply increasing model size?*

- ◯ Because larger models are more prone to overfitting with the same data
- ◯ Because performance improvements follow a power-law, making additional gains progressively smaller
- ◯ Because increased model size always necessitates a proportional exponential increase in data filtering
- ◯ Because the training loss suddenly plateaus after a specific size is reached

Because performance improvements follow a power-law, making additional gains progressively smaller

13.7. **Short answer:** What was the key insight related to the development of the Phi family of small LLMs?

Data quality is an underrated factor. Prioritize this over model size/amt of data. "Textbooks are all you need" - training on high quality data.