CSE 493 G1 / 599 G1
Deep Learning
Winter 2026 Exam 2

March 5, 2026

Full Name (as on Gradescope): ______________________________

UW Net ID: ______________________________

Question		Score
True/False	(30 pts)	
Multiple Choice	(32 pts)	
Free Response 1	(18 pts)	
Free Response 2	(20 pts)	
Total	(100 pts)	

Welcome to the CSE 493 G1 / 599 G1 Exam!

- The exam is 80 min and is **double-sided**.
- You may use a non-graphing calculator and no other electronic devices.
- One handwritten double sided cheat sheet is allowed.

I understand and agree to uphold the University of Washington Student Conduct Code during this exam.

Signature: ______________________ Date: ______________

Good luck!

This page is left blank for scratch work only. DO NOT write your answers here.

This page is left blank for scratch work only. DO NOT write your answers here.

This page is left blank for scratch work only. DO NOT write your answers here.

1 True/False (30 points) - Recommended 15 Minutes

Fill in the circle next to your choice (like this: ●). No explanations are required.

Each question is worth 3 points

1.1 Two 3×3 convolutions with D input and D output channels have more parameters than one 5×5 convolution with D input and D output channels.

○ True

○ False

1.2 Two models with the same number of parameters but different architectures will require roughly the same amount of memory to perform a forward and backward pass on the same data.

○ True

○ False

1.3 Generally, larger vocabulary sizes for tokenizers will lead to shorter sequence lengths.

○ True

○ False

1.4 The number of parameters in an RNN increases linearly with the sequence length.

○ True

○ False

1.5 Increasing the number of attention heads in a multi-head attention layer increases the total number of attention maps computed, since each head produces its own separate attention map.

○ True

○ False

1.6 Imagine you are training a language model that consists of a single stack of attention and MLP blocks, rather than an encoder-decoder. You should use unmasked (bidirectional) attention during training so the model can see the full context, and then switch to causal (masked) attention at generation time so the model only attends to previously generated tokens.

○ True

○ False

1.7 If you want a network to produce two different types of outputs simultaneously, such as a probability distribution over class labels and bounding box coordinates, you can achieve this by attaching multiple separate fully connected heads to the same shared feature representation, each with its own output dimension and loss function

- ◯ True
- ◯ False

1.8 A key idea in contrastive self-supervised learning (e.g., SimCLR) is that two augmented views of the same image should have similar representations, while views of different images should be pushed apart.

- ◯ True
- ◯ False

1.9 Transfer learning only works when the source task and target task have the same set of output classes, because the learned features are specific to the classes the network was originally trained on

- ◯ True
- ◯ False

1.10 Self-supervised learning models are <u>defined</u> by the fact that the model's output is a similarity score between embeddings or a reconstructed input, rather than a probability distribution over discrete class labels.

- ◯ True
- ◯ False

2 Multiple Choice (32 points) - Recommended 25 Minutes

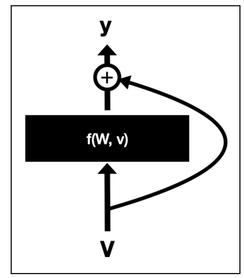
Fill in the circle next to the letter(s) of your choice (like this: ●). No explanations are required. **Choose ALL options that apply.**

Each question is worth 8 points and the answer may contain multiple correct options, or none at all. Selecting all of the correct options and none of the incorrect options will get full credit. For questions with multiple correct options, each incorrect or missing selection gets a 4-point deduction (up to 8 points).

2.1 You are training a neural network. At some point in the network, there is an activation vector $\mathbf{v}$ of shape $(1 \times d)$. This vector is fed into a linear layer f with weight matrix $\mathbf{W}$ of shape $(d \times d)$ (i.e., $f(\mathbf{v}) = \mathbf{vW}$). There is a residual connection from just before f to just after f, as shown to the right.
The output is:

$$\mathbf{y} = f(\mathbf{v}) + \mathbf{v} = \mathbf{vW} + \mathbf{v}$$

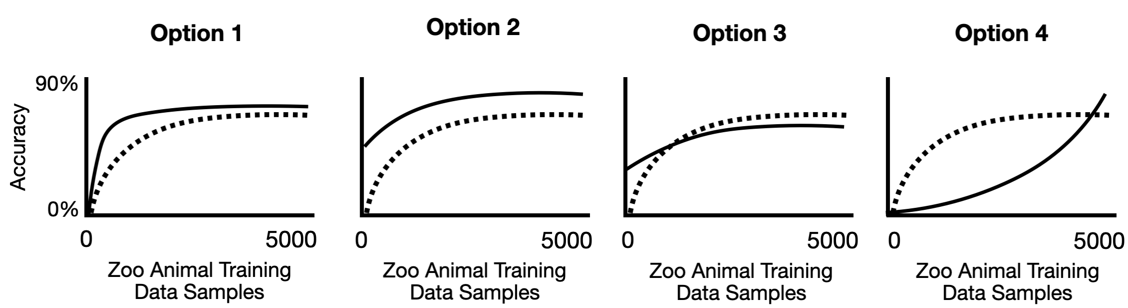
Assume that $\frac{\partial \mathcal{L}}{\partial \mathbf{y}}$ is given (shape $1 \times d$). Let $\mathbf{I}$ denote the identity matrix. What is $\frac{\partial \mathcal{L}}{\partial \mathbf{v}}$?



◯ A: $\frac{\partial \mathcal{L}}{\partial \mathbf{y}}\mathbf{W}^T$
◯ B: $\frac{\partial \mathcal{L}}{\partial \mathbf{y}}\mathbf{W}^T + \mathbf{I}$
◯ C: $\frac{\partial \mathcal{L}}{\partial \mathbf{y}}(\mathbf{I} + \mathbf{W}^T)$
◯ D: $(\mathbf{I} + \frac{\partial \mathcal{L}}{\partial \mathbf{y}})\mathbf{W}^T$

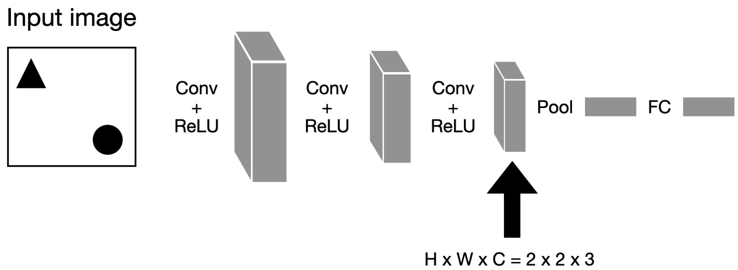
2.2 You pretrain a convolutional neural network on ImageNet. You then transfer it to a smaller Zoo Animal classification dataset with 500 classes by removing the final fully-connected classification layer and replacing it with a new, randomly initialized fully-connected layer for your target classes. You then fine-tune the entire network on your target data. The **dashed line** in each plot below shows the accuracy curve when training the same architecture from **random initialization** on the target dataset. The **solid line** shows the accuracy curve of the fine-tuned model.

Which plot best represents the **expected training curve** of the fine-tuned model compared to training from scratch? **Select one.**

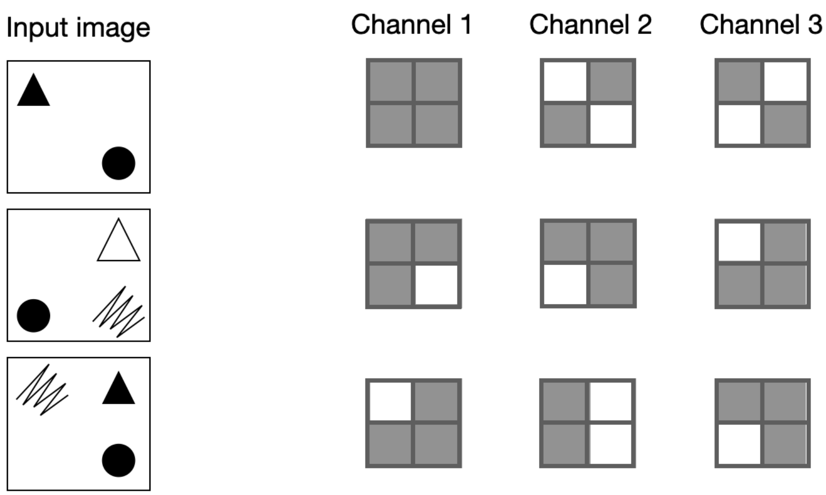


◯ A: Option 1
◯ B: Option 2
◯ C: Option 3
◯ D: Option 4

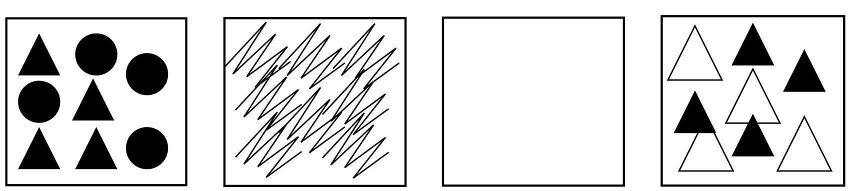
2.3 You have trained the CNN shown below and want to understand what features it has learned at a specific layer. You examine the activations at the layer indicated by the arrow, which has shape $H \times W \times C = 2 \times 2 \times 3$.



You pass three different input images through the network and visualize the resulting activations across all 3 channels. In the activation grids below, **white squares indicate high activation values** and **gray squares indicate low activation values**.



You then perform **gradient ascent** on a randomly initialized input image to maximize the sum of all 4 spatial values in **Channel 3** of this activation. Which of the following images would you most expect gradient ascent to produce? **Select one.**



Option 1 Option 2 Option 3 Option 4

- ◯ A: Option 1
- ◯ B: Option 2
- ◯ C: Option 3
- ◯ D: Option 4

2.4 For this question, use the following notation and assumptions:

- Let $T(f(x))$ denote the time it takes to compute $f(x)$
- Let $\text{Encode}_m(n)$ denote the process of encoding a sequence of length n into a vector representation using model m (e.g., converting a user's input question into a hidden representation for later processing)
- Let $\text{Decode}_m(n)$ denote the process of autoregressively generating n new tokens using model m (e.g., generating a response to a user's question, or captioning an unseen image)
- Let tf denote a Transformer and let rnn denote an RNN

Assume we have **infinite compute and memory**, so any operations that can be parallelized are fully parallelized. Under this assumption:

- Independent operations run in parallel: $T(f(x) + g(x)) = \max\big(T(f(x)),\ T(g(x))\big)$
- Sequential dependencies still require waiting: $T(g(f(x))) \approx T(f(x)) + T(g(y))$ where $y = f(x)$

Select all statements that are TRUE:

○ A: $T(\text{Encode}_{tf}(n)) \approx T(\text{Encode}_{tf}(2n))$

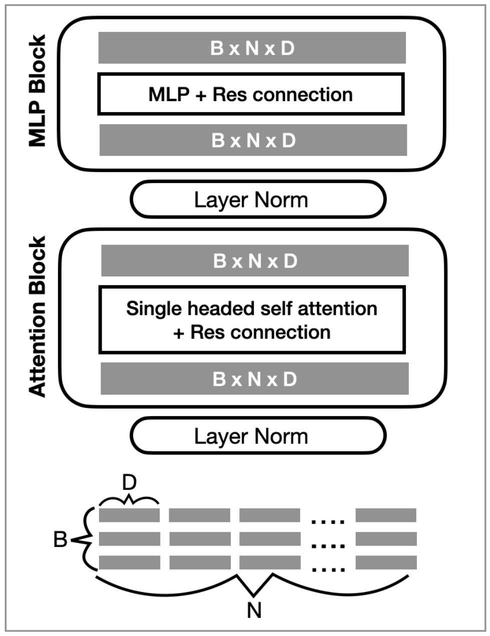
○ B: $T(\text{Encode}_{rnn}(n)) \approx T(\text{Encode}_{rnn}(2n))$

○ C: $T(\text{Decode}_{tf}(n)) \approx T(\text{Decode}_{tf}(2n))$

○ D: $T(\text{Decode}_{rnn}(n)) \approx T(\text{Decode}_{rnn}(2n))$

3 Transformer Memory (18 points) - Recommended 15 Minutes

You are training a Transformer that consists of n layers, where each layer contains a single-head self-attention block followed by an MLP block, as shown below. The MLP block consists of two linear layers with a nonlinearity in between. The input and output dimensions of both linear layers are equal and of size D. You perform a forward pass on a batch of sequences. A single element of a sequence within a batch is a vector of dimension D. So the input/output of each attention and MLP block has shape $B \times N \times D$, where B is the batch size, N is the sequence length, and D is the embedding dimension.



Assume that N is approximately $100\times$ larger than both B and D (i.e., $N \gg B$ and $N \gg D$). *Ignore the memory required to store model weights—consider only the memory for intermediate values stored during the forward pass (e.g., activations, attention maps).* For each of the following scenarios, write the **approximate multiplicative factor** by which the total memory stored during the forward pass changes. Your answer should reflect the **dominant** memory term—you may ignore terms that are negligibly small relative to the largest term given our assumptions. Write **1x** if the memory does not meaningfully change. You are not using any efficient attention implementations, such as Flash Attention.

3.1 Memory in the Attention Block (3 pts each)

a) The **batch size B** doubles. **Attention block** memory increases by approximately $\boxed{}$ x

b) The **sequence length N** doubles. **Attention block** memory increases by approximately $\boxed{}$ x

c) The **embedding dimension** D doubles. **Attention block** memory increases by approximately $\boxed{}$ x

3.2 Memory in the MLP Block (3 pts each)

a) The **batch size** B doubles. **MLP block** memory increases by approximately $\boxed{}$ x

b) The **sequence length** N doubles. **MLP block** memory increases by approximately $\boxed{}$ x

c) The **embedding dimension** D doubles. **MLP block** memory increases by approximately $\boxed{}$ x

4 Transformer Arithmetic (20 points) - Recommended 20 Minutes

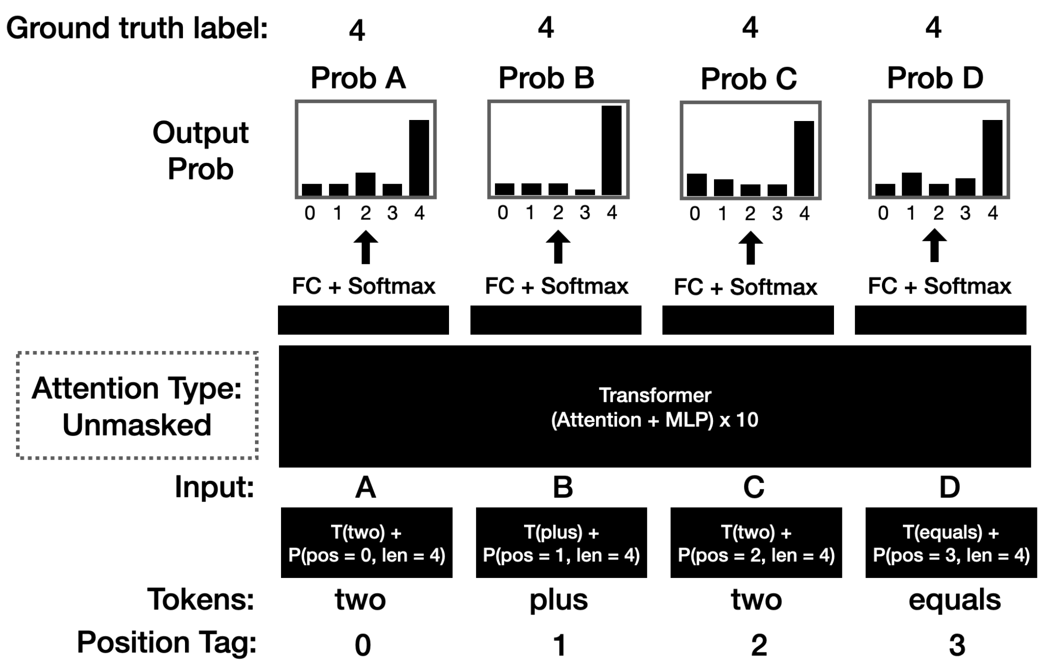
You have trained a Transformer to perform simple arithmetic in a supervised fashion. The training data consists of text sequences paired with numerical ground truth labels. For example:

Input sequence	Ground truth label
One Plus Two Equals	3
Two Minus One Equals	1
One Plus One Plus One Equals	3

The input vocabulary is {One, Two, Plus, Minus, Equals} and the output classes are {0, 1, 2, 3, 4}. The final token in every sequence is always "Equals".

The model consists of 10 stacked blocks of (Attention + MLP). Each input token is represented as the sum of a token embedding and a positional embedding. The positional encoding encodes both the position of each token and the total length of the sequence.

At each input position, the model produces a probability distribution over the output classes using an FC layer + Softmax. The same ground truth label is used as the target at **every** output position, and the model is trained with cross-entropy loss. The setup is illustrated below for the input "Two Plus Two Equals" (ground truth label: 4):



For each of the following scenarios, you are given an input sequence and an attention type (either unmasked or masked causal). Assume the model was trained and is evaluated using the attention type shown. Your task is to sketch the approximate probability distribution you would expect at the specified output position. If the model can confidently determine the answer, draw a single tall bar at that value. If the model is uncertain, draw bars reflecting the plausible outputs. **Read each table carefully, pay close attention to the attention type, the input tokens, their positions, and which output position you are asked about.**

Here is an example of a table specifying a certain scenario and what we want from you.

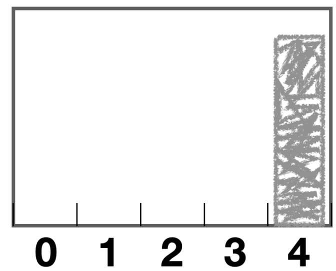
Read this table carefully

Example of what we want from you

Pay attention to which output we want!

Attention	Unmasked			
Input	A	B	C	D
Token	Two	Plus	Two	Equals
Position	0	1	2	3
Seq Len	4	4	4	4
Which output	★			

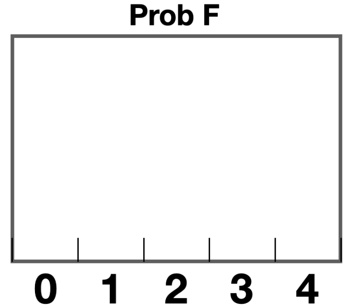
Prob A



4.1 Prob F given the below input and settings (4 pts)

Attention	Unmasked					
Input	A	B	C	D	E	F
Token	One	Plus	One	Plus	One	Equals
Position	0	1	2	3	4	5
Seq Len	6	6	6	6	6	6
Which output						★

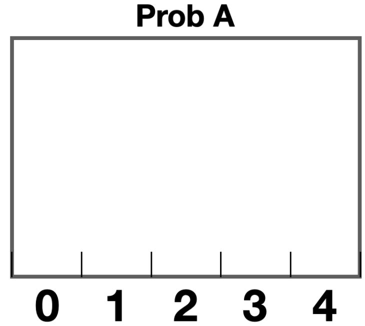
Prob F



4.2 Prob A given the below input and settings (4 pts)

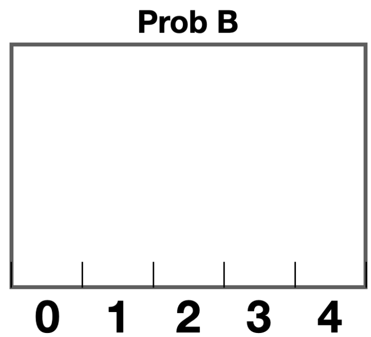
Attention	Unmasked					
Input	A	B	C	D	E	F
Token	One	Plus	One	Plus	One	Equals
Position	0	1	2	3	4	5
Seq Len	6	6	6	6	6	6
Which output	★					

Prob A



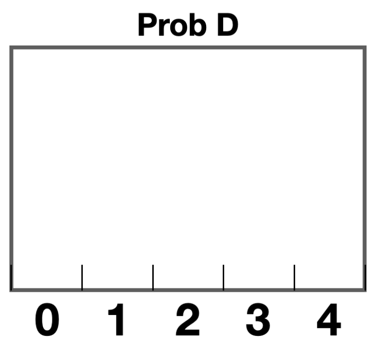
4.3 Prob B given the below input and settings (4 pts)

Attention	Masked Causal			
Input	A	B	C	D
Token	Two	Plus	One	Equals
Position	0	1	2	3
Seq Len	4	4	4	4
Which output		★		



4.4 Prob D given the below input and settings (4 pts)

Attention	Masked Causal			
Input	A	B	C	D
Token	Two	Plus	One	Equals
Position	0	1	2	3
Seq Len	4	4	4	4
Which output				★



4.5 Prob B given the below input and settings (4 pts)

Attention	Unmasked					
Input	A	B	C	D	E	F
Token	One	Plus	One	Minus	Two	Equals
Position	4	1	2	3	0	5
Seq Len	6	6	6	6	6	6
Which output		★