Quiz 6 - Generalization

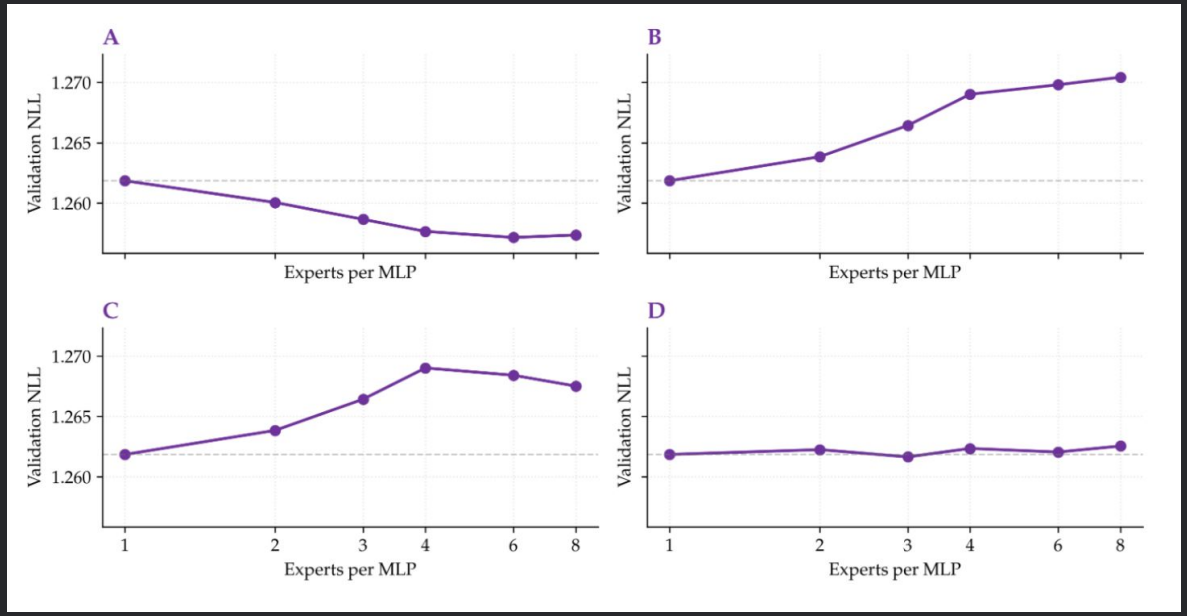
11/19

DNA + Generalization retrospective

- Absolute loss diffs are smaller, loss is much higher variance (quiz answers have seed replication)
- Generalization has a lot of interaction with previous problem material
- Otherwise, hopefully felt a lot like standard language model pre-training
- Many of the assignment/quiz problems are unknown to literature. Be proud of yourself for whatever happens!

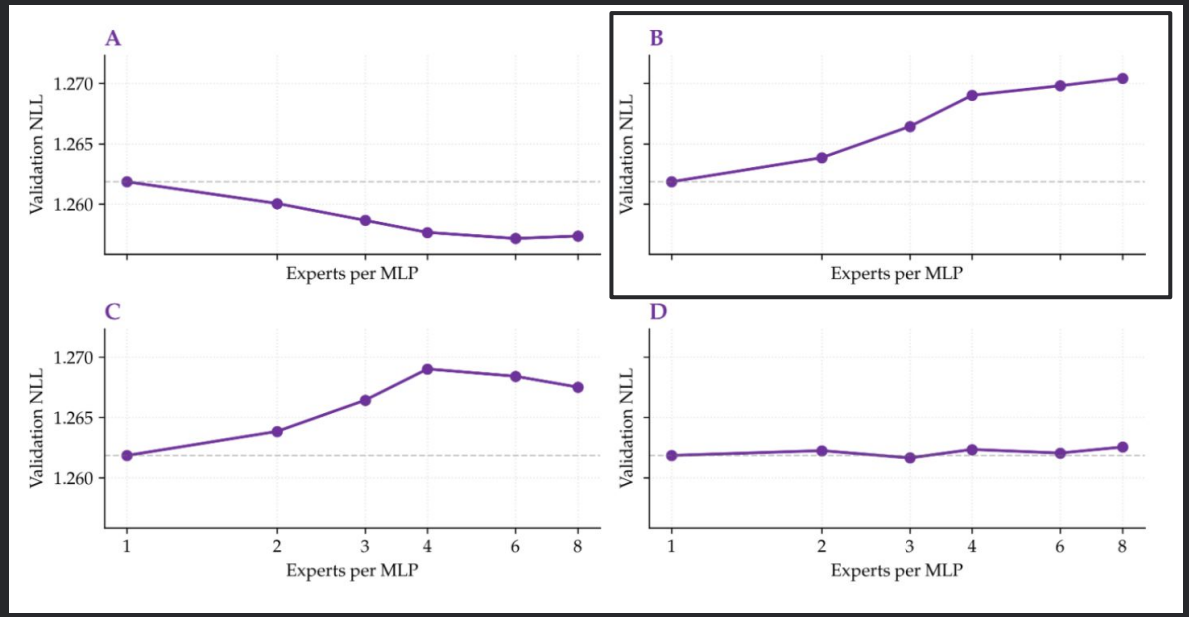
Problem 1 - Architecture

1a: Which plot best captures loss as a function of expert count?



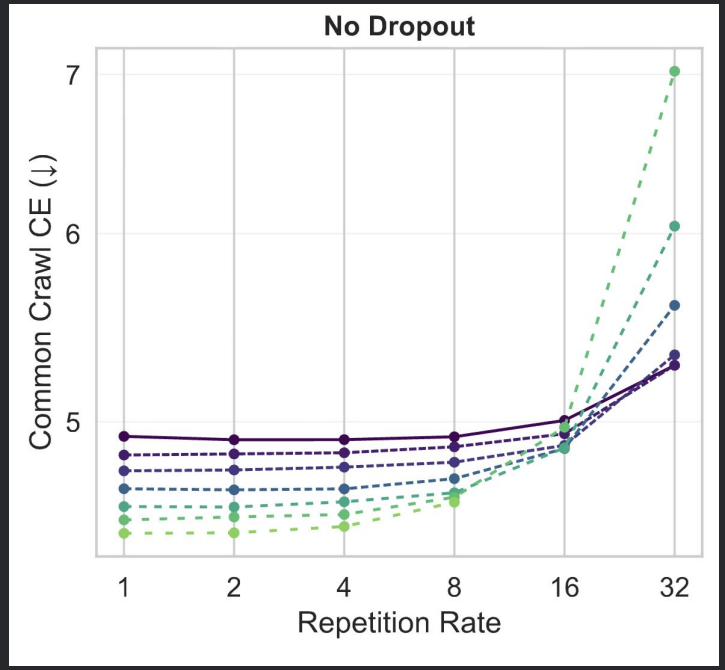
Problem 1 - Architecture

Answer: Option B



Problem 1 - Architecture

Reasoning: As discussed in lecture, MoE's with no additional regularization have trouble in data-constrained regime due to overfitting



Problem 1 - Architecture

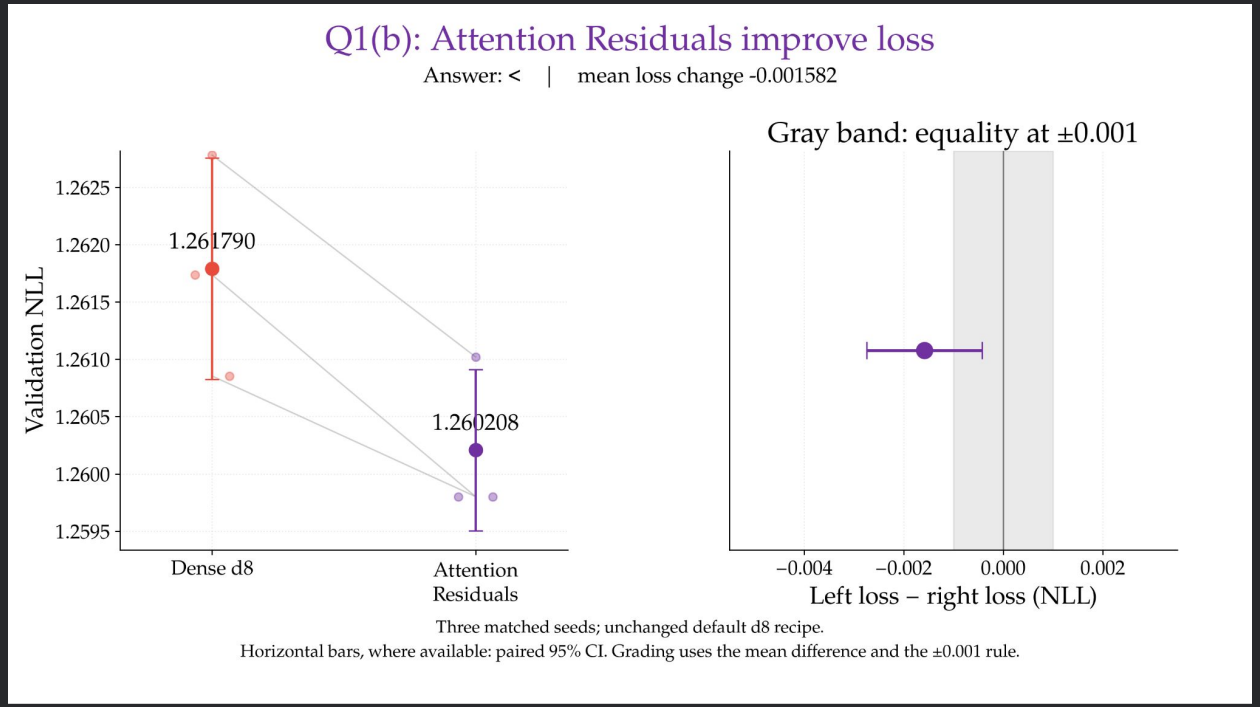
1b: Does attention residuals improve, hurt, or maintain the loss?

(b) Attention Residuals (6 points). Replace ordinary residual sums with learned softmax attention over earlier attention/MLP outputs and the embedding (Full Attention Residuals). At the default d8 recipe, does this give lower, similar, or higher validation loss than ordinary dense d8? Write $=$ for losses within 0.001 NLL (estimated run-to-run standard deviation), and $<$ or $>$ otherwise.

L_{AttnRes} ☐ L_{dense}

Problem 1 - Architecture

Answer: $L_{AttnRes} < L_{dense}$



Problem 1 - Architecture

Reasoning: For architecture, interventions that don’t affect description length or program runtime generally transfer between compute and data-efficient regimes. For example, Transformer is likely better than RNN in both

Problem 2 - Stochastic regularization

2ai + 2aii: Does dropout improve a {single model, 10 ensemble}

(a) Attention dropout (12 points). Use probability 0.05, with no stochastic depth.

(i) Single model (4 points). Compare dropout with no dropout.

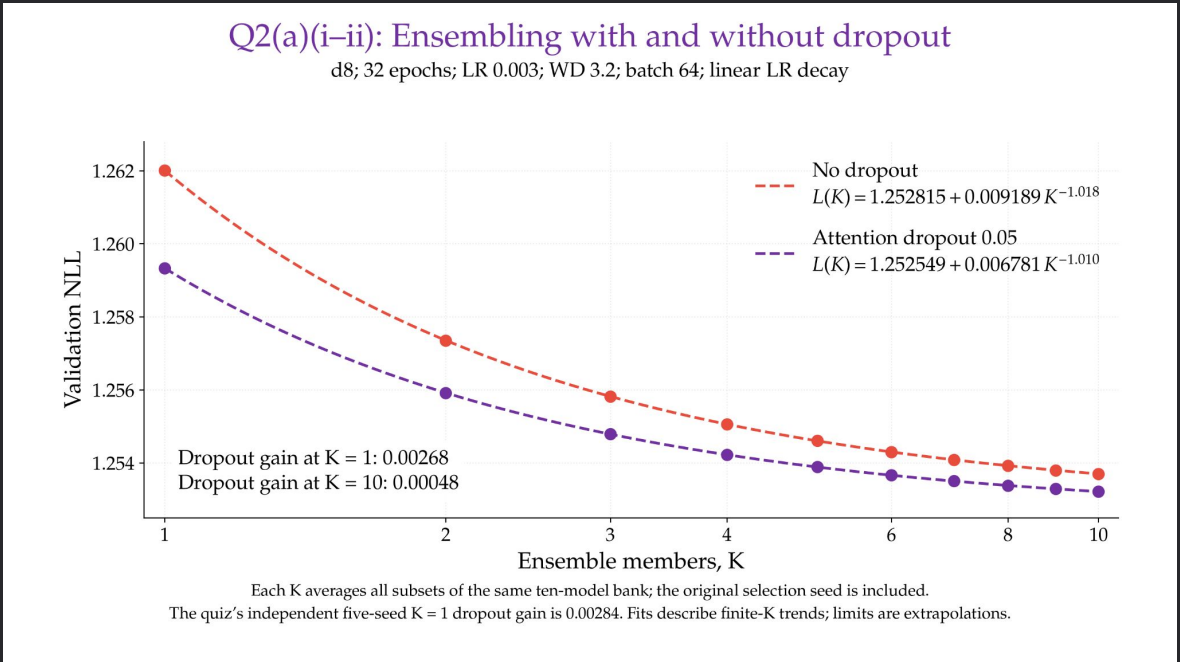
$L_{\text{dropout, K=1}}$ ☐ $L_{\text{no dropout, K=1}}$

(ii) Ten-model ensemble (4 points). Compare ensembles trained with and without dropout.

$L_{\text{dropout, K=10}}$ ☐ $L_{\text{no dropout, K=10}}$

Problem 2 - Stochastic regularization

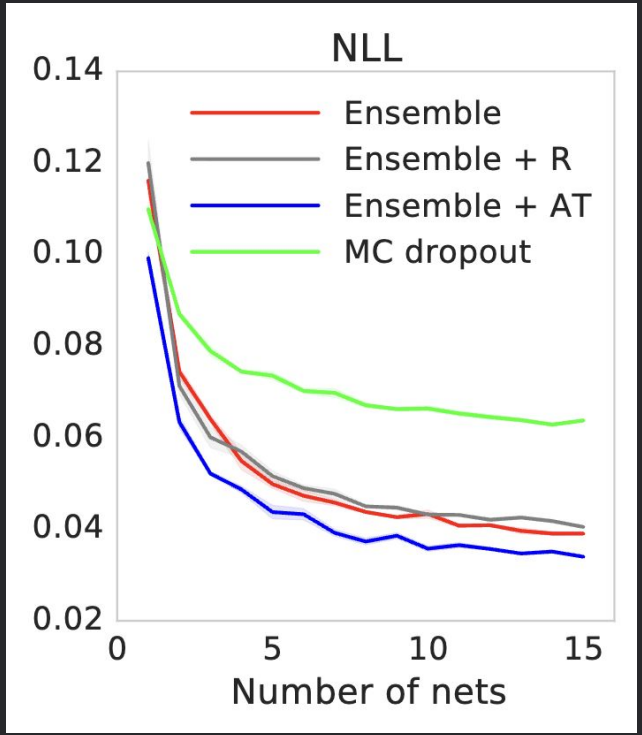
Answer: Dropout improves a single model but not an ensemble



Problem 2 - Stochastic regularization

Reasoning: Dropout and ensembling contribute to the same form of regularization (in fact, they were motivated as such) and are unlikely to compose.

Empirically seen that dropout performs a worse version of ensembling [Lakshminarayanan et al, 2016]



Problem 2 - Stochastic regularization

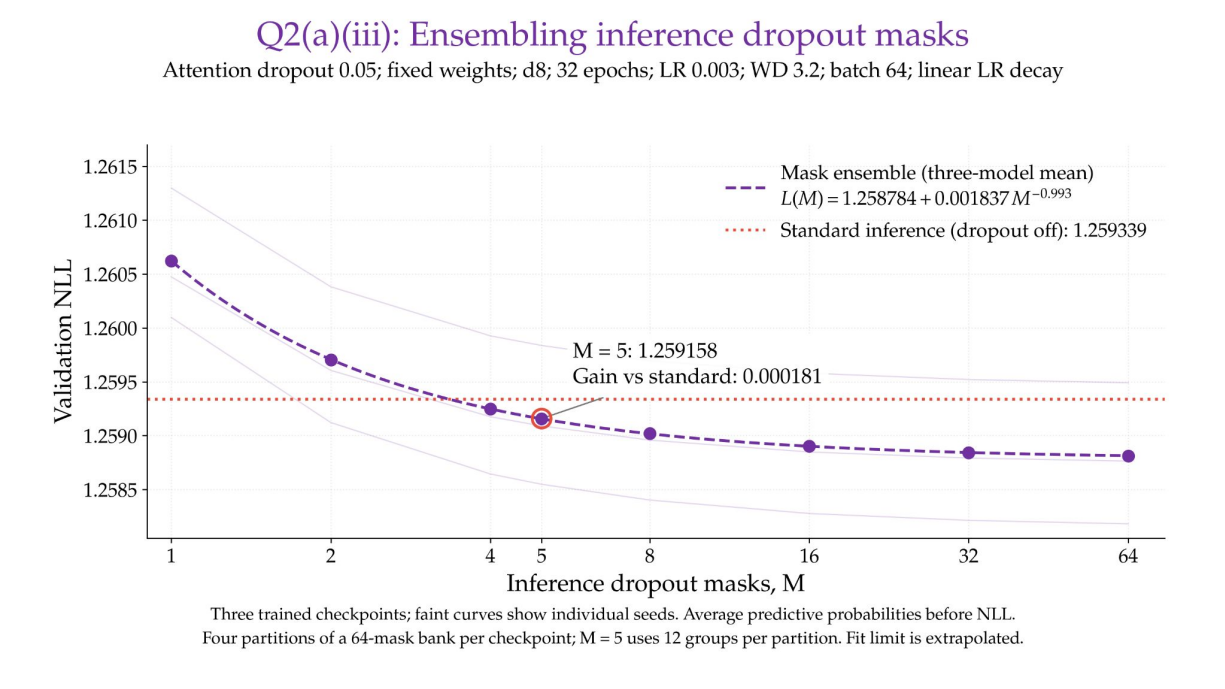
1aiii: Can you ensemble dropout masks for better loss?

(iii) Inference masks (4 points). For one dropout-trained model, compare standard inference (dropout off) with averaging prediction probabilities over five independent dropout masks (dropout on). Keep the weights fixed.

$L_{\text{five masks}}$ ☐ L_{standard}

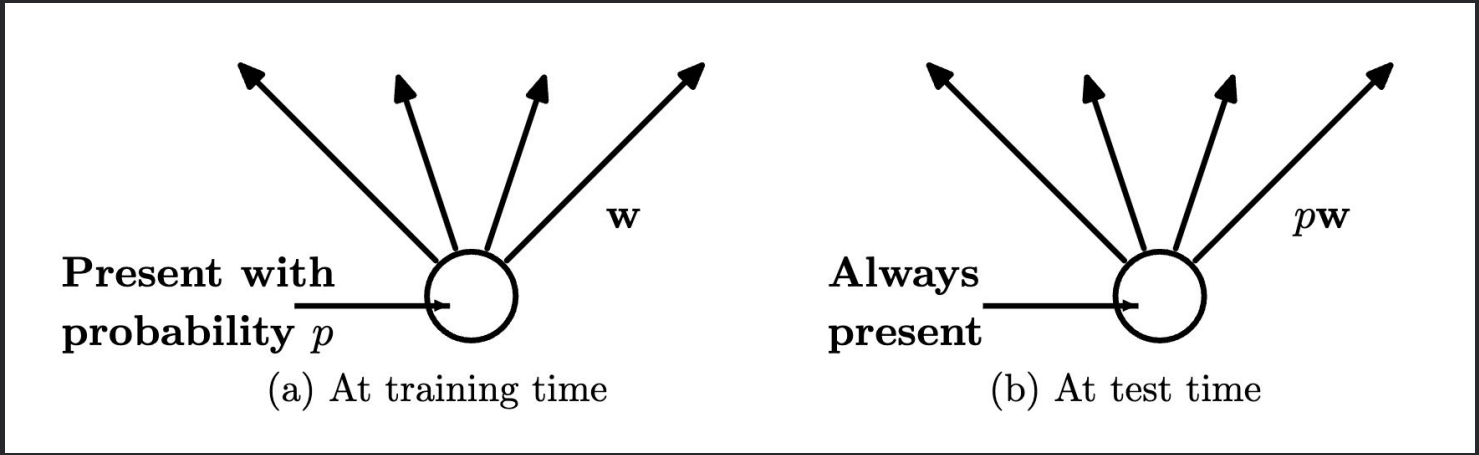
Problem 2 - Stochastic regularization

Answer: No, =



Problem 2 - Stochastic regularization

Reasoning: Dropout at inference time is intuitively the same as weight average of all potential masks. As long as all of the masks are in the same basin, dropout is already aggregating all of the solutions



Problem 2 - Stochastic regularization

2b: Does stochastic depth compose with dropout or ensembling?

(b) Stochastic depth (8 points). Stochastic depth (SD) randomly skips residual branches during training:

$$h \leftarrow h + \frac{m}{1 - p_i} f(h), \qquad m = \begin{cases} 0 & \text{with probability } p_i, \\ 1 & \text{otherwise.} \end{cases}$$

Here f is an attention or MLP residual branch; rescaling preserves its expected contribution. For the six interior block applications, use $p_i = p_{\max} i/6$. Sample masks independently per sequence, branch, and application (including loop repeats), shared across tokens and channels. Keep the first and last blocks active. At evaluation use $h \leftarrow h + f(h)$, without masking or rescaling.

At $p_{\max} = 0.05$, SD improves dense d8 loss by about 0.0018. Does this benefit compose with:

(i) Attention dropout 0.05 **(4 points).**

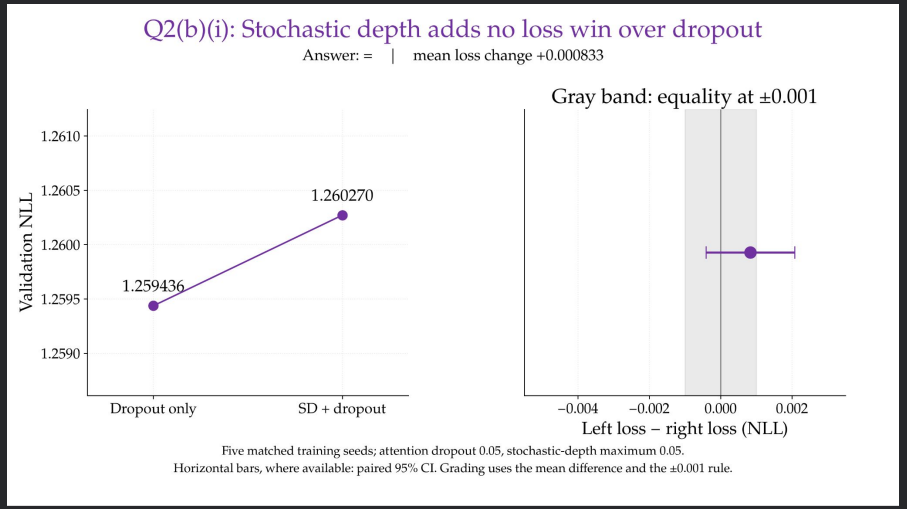
$L_{\text{SD+dropout}}$ ☐ L_{dropout}

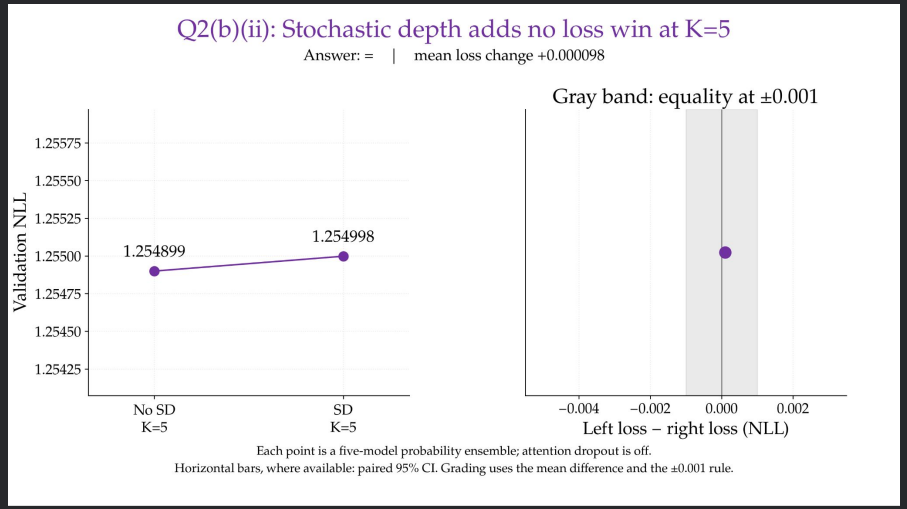
(ii) Five-model ensembling, without dropout (4 points).

$L_{\text{SD, K=5}}$ ☐ $L_{\text{no SD, K=5}}$

Problem 2 - Stochastic regularization

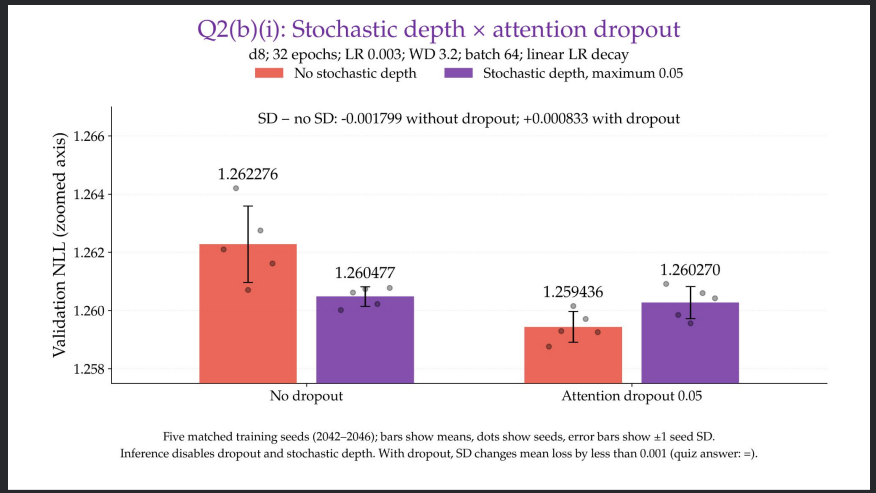
Answer: Neither, =, =

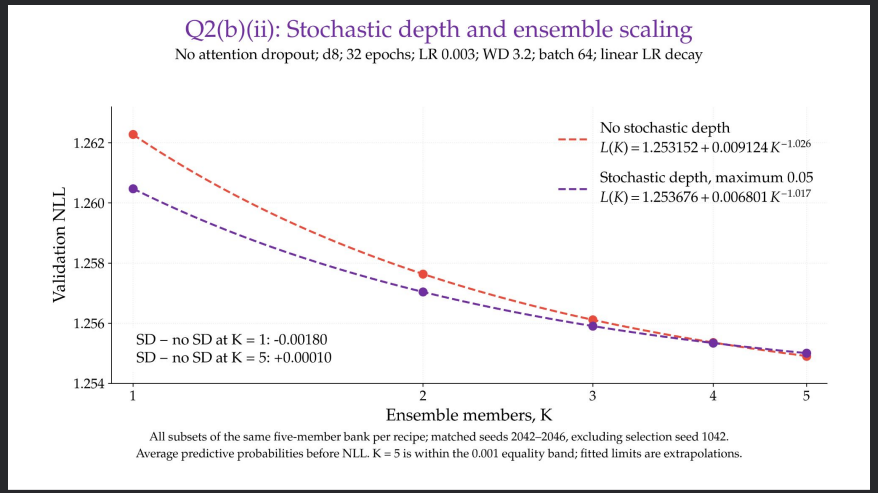




Problem 2 - Stochastic regularization

Reasoning: Stochastic depth is a dropout-like augmentation. Similar to how dropout doesn't compose strongly with ensembling, stochastic depth doesn't additionally compose with dropout and ensembling





Problem 3 - Fine-tuning

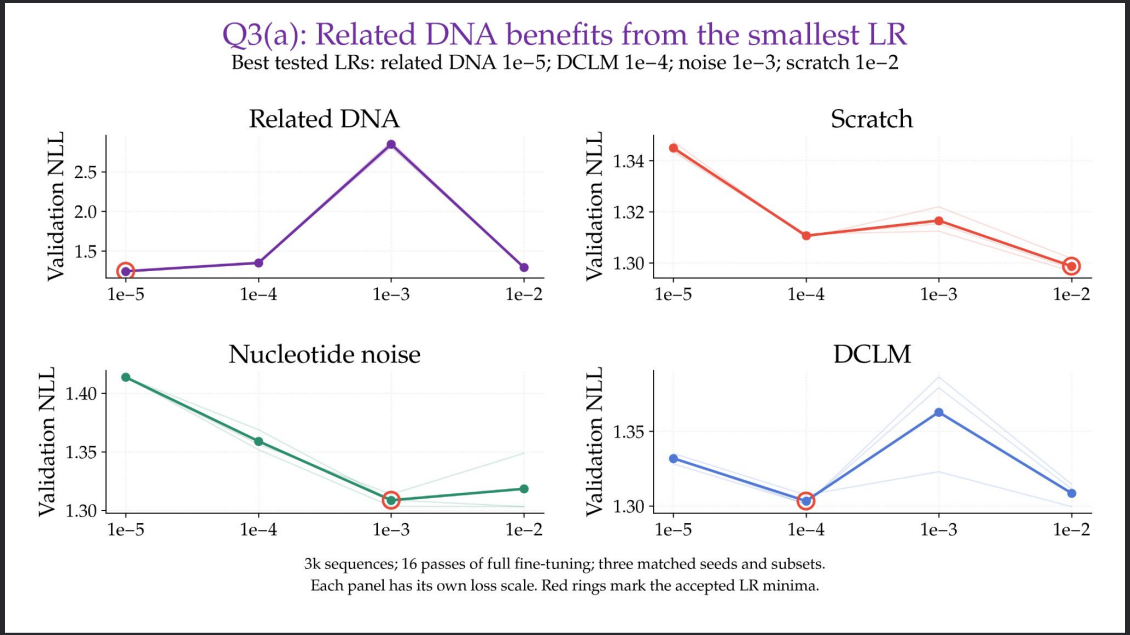
3a: Which pre-trained initialization benefits from the lowest learning rate?

(a) Fine-tuning LR (6 points). At 3k samples, which initialization benefits from the smallest LR? Compare each initialization's best LR over $\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$ and select the one with the smallest loss-minimizing LR.

Related DNA ☐ Scratch ☐ Nucleotide noise ☐ DCLM ☐

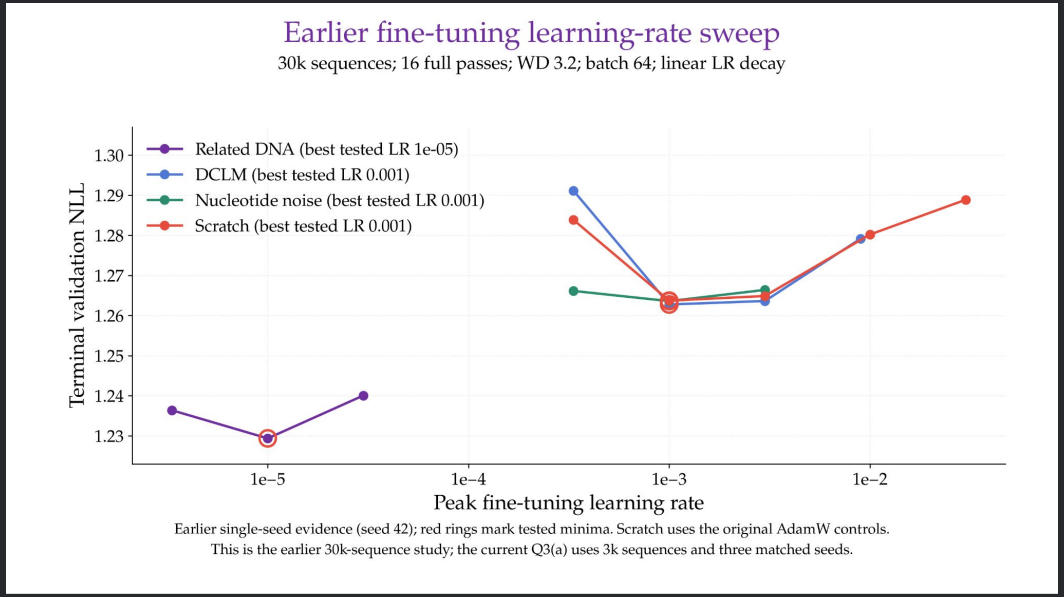
Problem 3 - Fine-tuning

Answer: Related DNA pre-training



Problem 3 - Fine-tuning

Reasoning: The related DNA pre-training has the most useful features for the fine-tuning task. Intuitively, larger learning rate might be more damaging to helpful features from pre-training



Problem 3 - Fine-tuning

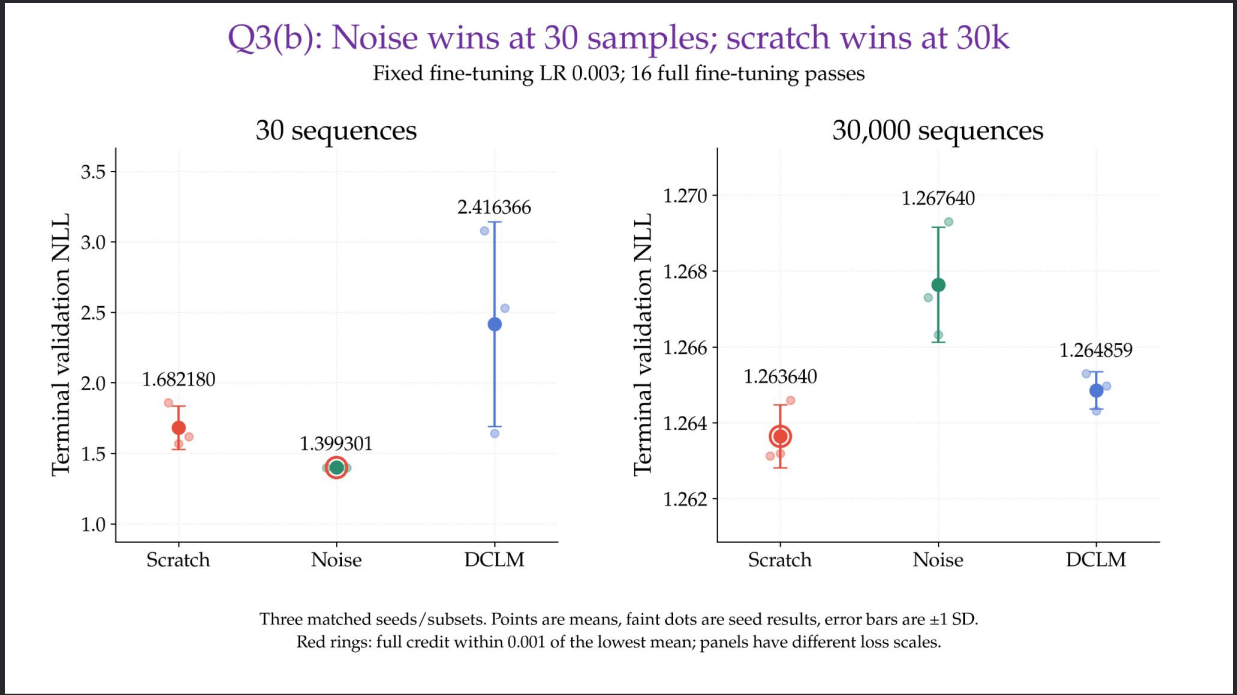
3b: What is the best initialization for 30 samples and 30,000 samples?

(b) Initialization (6 points). At LR 0.003, which initialization is best at each sample count? Each row is worth 3 points.

Samples	Scratch	Nucleotide noise	DCLM
30	☐	☐	☐
30,000	☐	☐	☐

Problem 3 - Fine-tuning

Answer: 30 samples prefers nucleotide noise, 30,000 samples prefers scratch



Problem 3 - Fine-tuning

Reasoning: At zero samples, nucleotide noise is best because it gets the frequency of tokens correct, surpassing the other models. However, with more fine-tuning, preferable to start from scratch since both sets of features are useless

Lesson: Lower pre-training loss doesn’t imply lower fine-tuning loss when unmatched

Problem 4 - Learning rate schedules

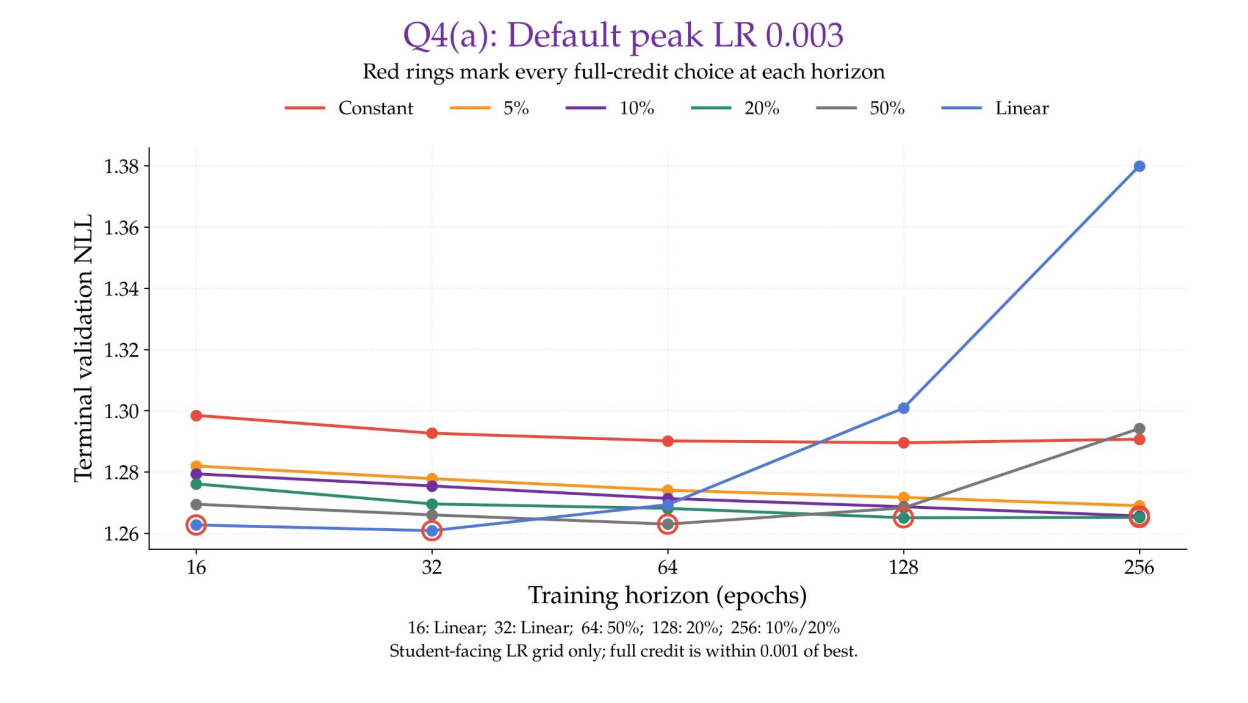
4a: At the default LR, which schedule should we use at each epoch count?

(a) Default LR (10 points). Fix peak LR to 0.003. Which schedule is best at each horizon?

Epochs	Constant	5% WSD	10% WSD	20% WSD	50% WSD	Linear
16	☐	☐	☐	☐	☐	☐
32	☐	☐	☐	☐	☐	☐
64	☐	☐	☐	☐	☐	☐
128	☐	☐	☐	☐	☐	☐
256	☐	☐	☐	☐	☐	☐

Problem 4 - Learning rate schedules

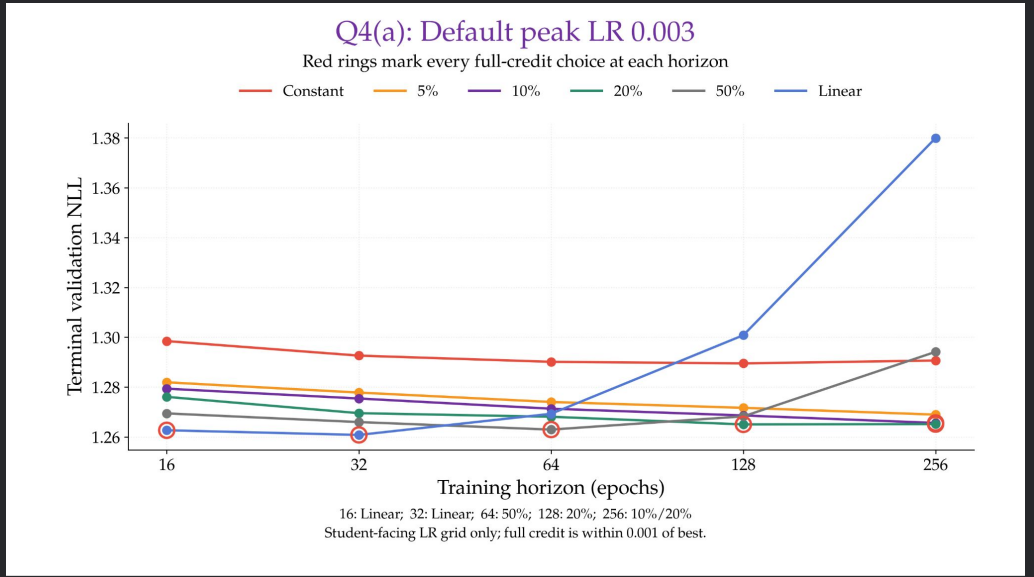
Answer: At the default LR, Linear -> Linear -> 50% -> 20% -> 20% (10%)



Problem 4 - Learning rate schedules

Reasoning: Constant and WSD require lower learning rates, bad for low epochs.

At higher epochs, linear overfits significantly but WSD has higher LR ⇒ lower sharpness ⇒ lower overfitting



Problem 4 - Learning rate schedules

4b: At tuned LR, which schedule should we use at each epoch count?

(b) Tuned LR (10 points). For each schedule and horizon, select the LR with the lowest three-seed mean over

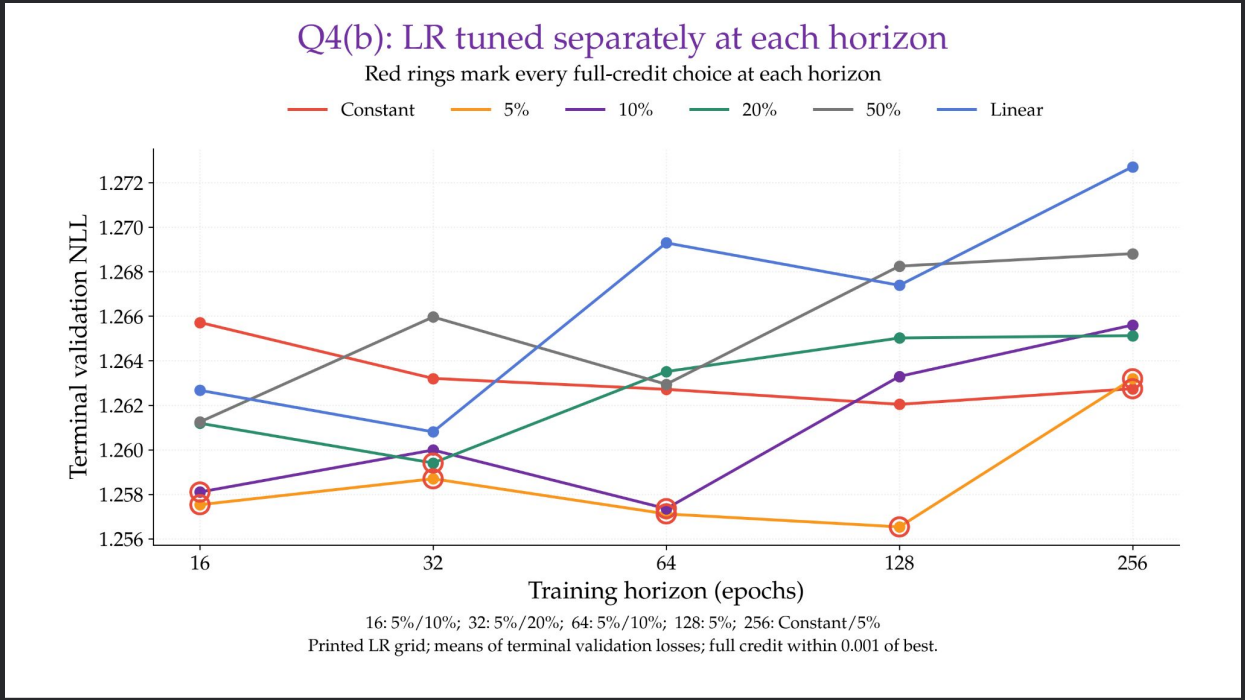
$$\{0.0001,\ 0.0003,\ 0.001,\ 0.003,\ 0.01\}.$$

Which schedule is best after tuning?

Epochs	Constant	5% WSD	10% WSD	20% WSD	50% WSD	Linear
16	☐	☐	☐	☐	☐	☐
32	☐	☐	☐	☐	☐	☐
64	☐	☐	☐	☐	☐	☐
128	☐	☐	☐	☐	☐	☐
256	☐	☐	☐	☐	☐	☐

Problem 4 - Learning rate schedules

Answer: 5% (10%) -> 5% (20%) -> 5% (10%) -> 5% -> Constant (5%)



Problem 4 - Learning rate schedules

Reasoning: After tuning learning rate, WSD can use smaller learning rates to do better

Q4(b): Chosen peak learning rates

Printed LR grid: 0.0001, 0.0003, 0.001, 0.003, 0.01

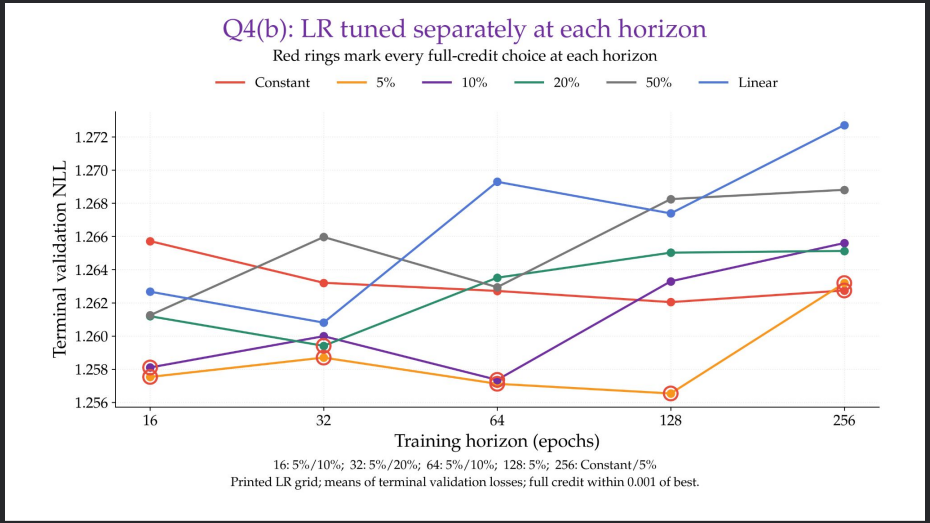
Epochs	Constant	WSD 5%	WSD 10%	WSD 20%	WSD 50%	Linear
16	0.0003	**0.0003**	**0.0003**	0.0003	0.001	0.003
32	0.0003	**0.0003**	0.001	**0.001**	0.003	0.003
64	0.0003	**0.001**	**0.001**	0.001	0.003	0.003
128	0.0003	**0.001**	0.001	0.003	0.003	0.01
256	**0.0003**	**0.001**	0.003	0.003	0.01	0.01*

Select the LR with the lowest completed-seed mean NLL per schedule/horizon. Purple: schedules within 0.001 of the row's best.
* Linear at 256 epochs is tentative (2/3 seeds at selected LR); other displayed LRs have three seeds. Excludes internal 1/3000 checks.

Problem 4 - Learning rate schedules

Reasoning: With tuned learning rate, constant can continue training at low sharpness without overfitting

However, we know that decay is helpful to get to the bottom of the valley. Therefore, minimal decay is good



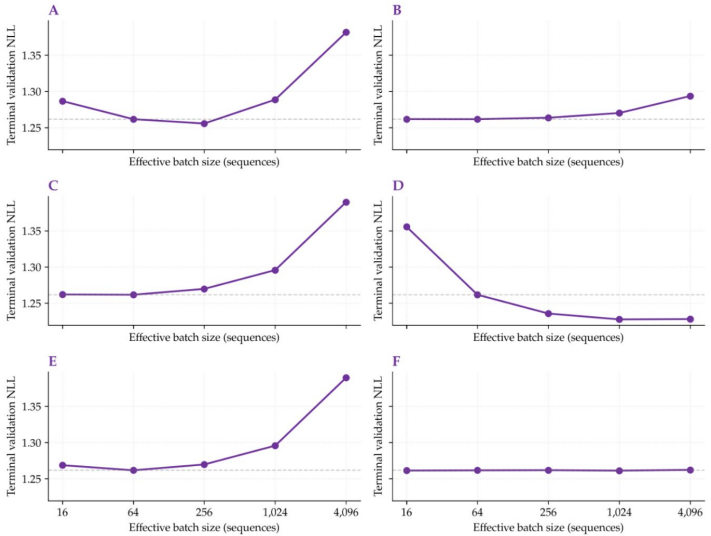
Problem 5 - Optimization

5a: Which of the following best captures batch size tradeoffs?

(a) Batch size (6 points). Vary $B \in \{16, 64, 256, 1024, 4096\}$ under the default 32-epoch linear schedule, transferring both LR and WD:

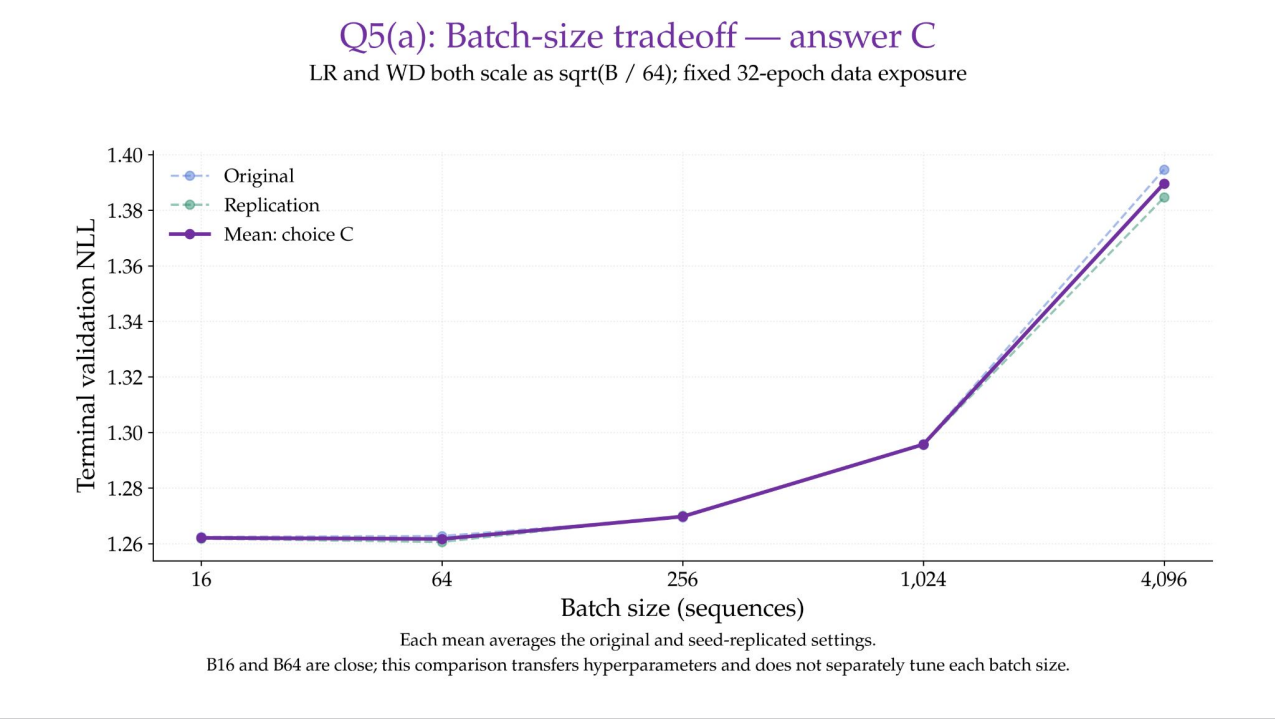
$$\mathrm{LR}(B) = 0.003\sqrt{B/64}, \qquad \mathrm{WD}(B) = 3.2\sqrt{B/64}.$$

Use $\lceil 4800/B \rceil$ warmup updates. Which plot matches mean terminal validation loss across the original and seed-replicated runs? Panels share axes and the $B = 64$ anchor.



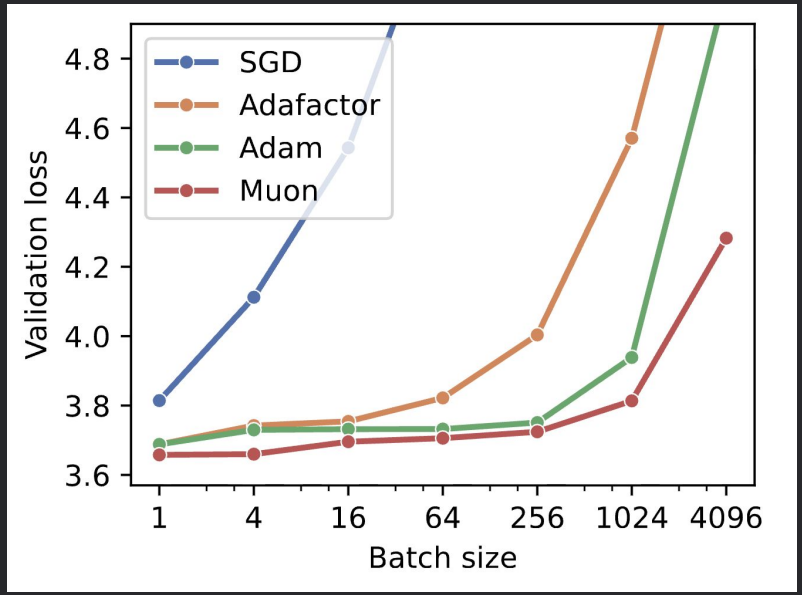
Problem 5 - Optimization

Answer: Option C



Problem 5 - Optimization

Reasoning: We know that reducing batch size while fixing tokens is generally a good intervention. A, C, E are all somewhat close. C better captures the fact that there's no regret from further reducing batch size.



Problem 5 - Optimization

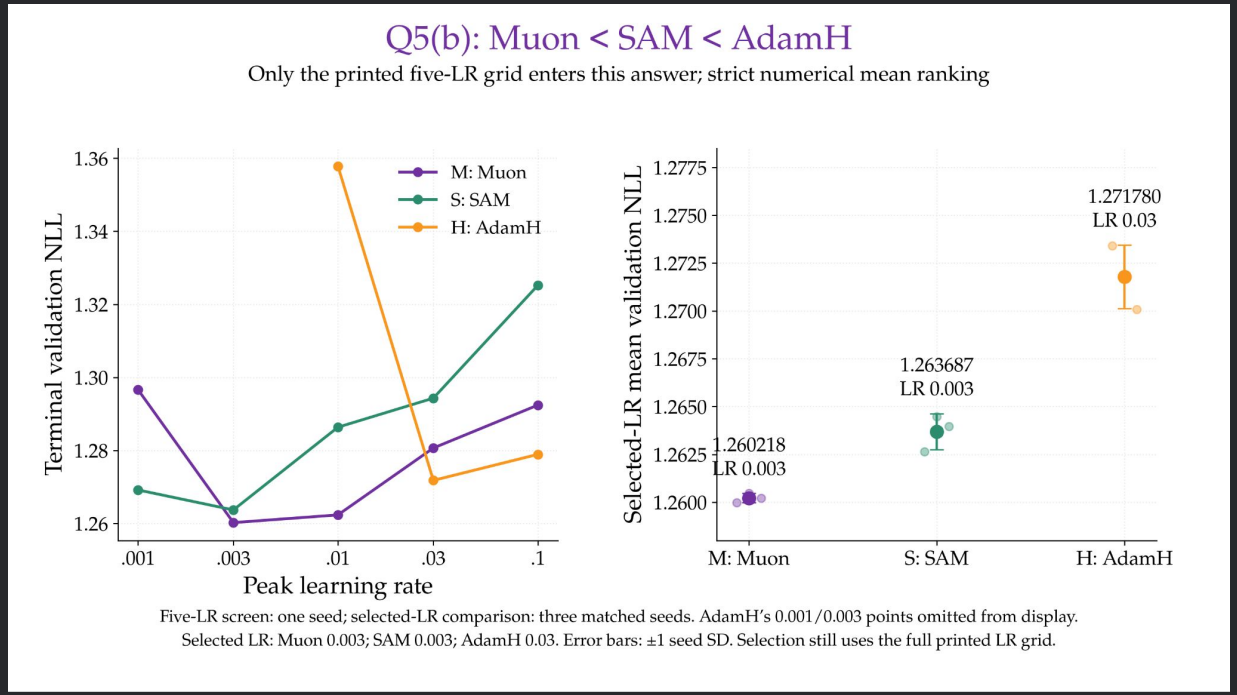
5b: What optimizer is best between Muon, SAM, and AdamH?

(b) Optimizers (6 points). Tune LR over $\{0.001, 0.003, 0.01, 0.03, 0.1\}$ for each optimizer. Then average terminal validation loss over three matched seeds and rank H = AdamH, M = Muon, S = SAM ($\rho = 0.05$). Keep the other defaults; AdamH uses no WD. Write letters and only $<$ or $>$, from lowest to highest mean loss. There is no equality option or tie rule.



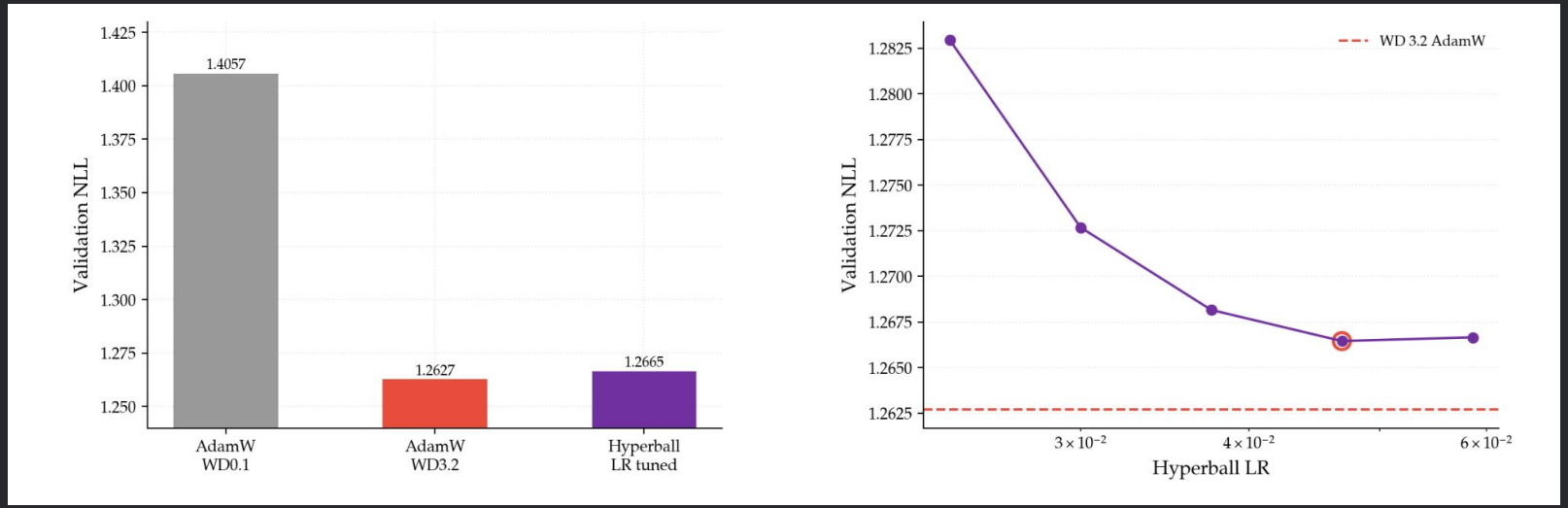
Problem 5 - Optimization

Answer: Muon < SAM < AdamH



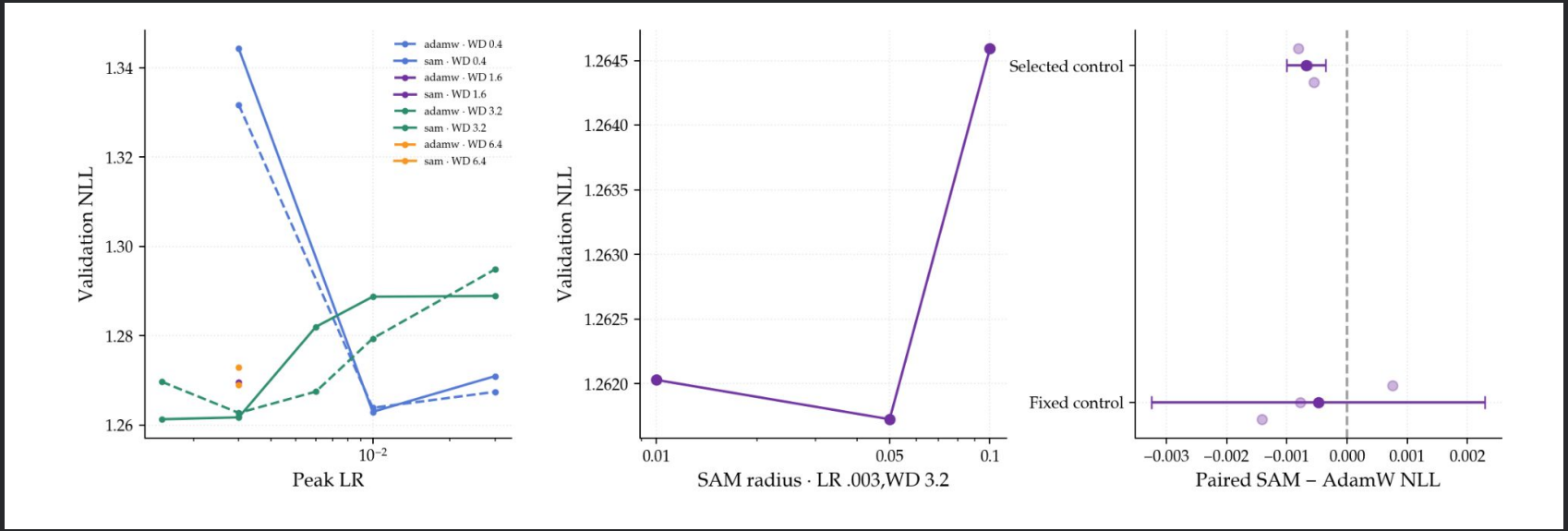
Problem 5 - Optimization

Reasoning: From assignment, you likely found that though AdamH obtains most of the benefit of tuning weight decay, it still lags behind AdamW baseline



Problem 5 - Optimization

Reasoning: From assignment, you likely found that SAM was comparable to AdamW (maybe slightly worse or slightly better depending on noise/scenario)



Problem 5 - Optimization

Reasoning: We generally know Muon is better than AdamW in compute-efficient regime. Similar to Attention Residuals, many compute-efficient improvements transfer to data-efficient regimes

Problem 6 - EMA

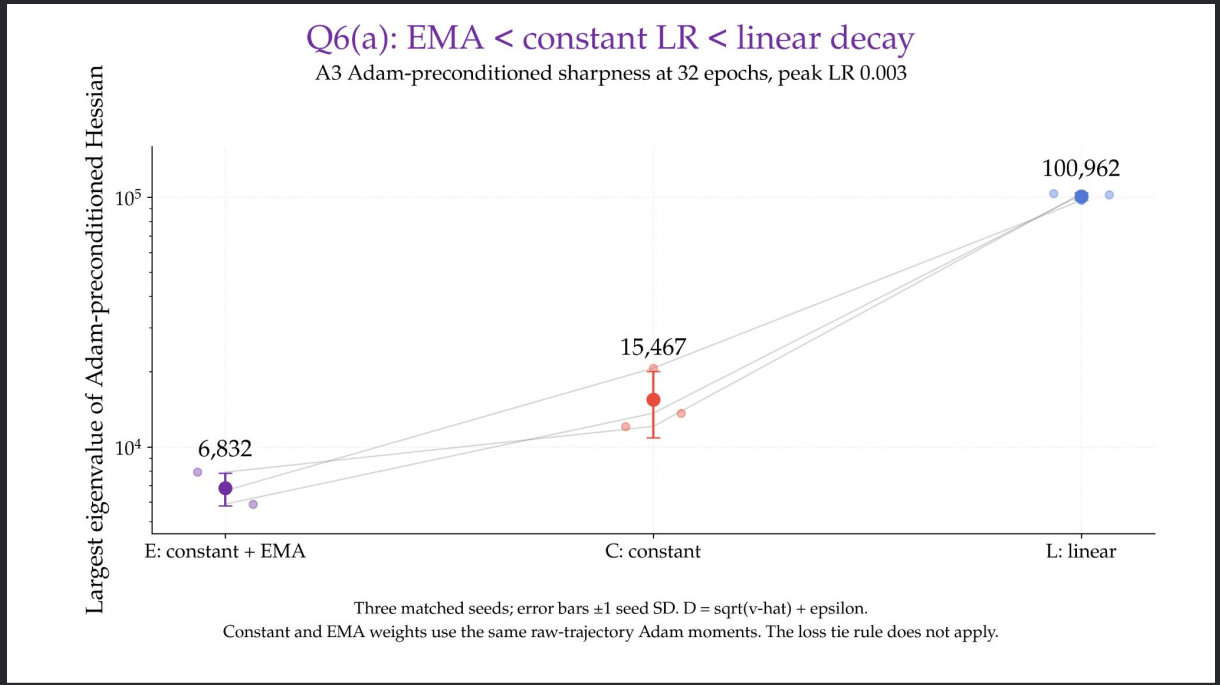
6a: What is the sharpness ranking between constant LR schedule, EMA, and linear?

(a) Sharpness (6 points). Fix LR to 0.003. Rank mean A3 Adam-preconditioned sharpness at 32 epochs: C = constant, E = constant with EMA, L = linear. This is the largest eigenvalue of $D^{-1/2}HD^{-1/2}$, where H is the loss Hessian and $D = \sqrt{\hat{v}} + \varepsilon_{\text{Adam}}$ uses saved Adam second moments. Use the constant trajectory's moments for both C and E. The 0.001 loss rule does not apply to sharpness.

$$\square \; < \; \square \; < \; \square$$

Problem 6 - EMA

Answer: EMA is flattest, constant is middle, learning rate decay is sharpest



Problem 6 - EMA

Reasoning: Intuitive that linear decay would increase sharpness since it lowers learning rate.

EMA is ***different*** from linear decay: It tracks the bottom of current region without entering a sharper region

Problem 6 - EMA

6b: What schedule and epoch count should we use?

(b) Loss with LR tuning (6 points). Peak LR is tuned over $\{0.0001, 0.0003, 0.001, 0.003\}$ separately for each recipe and epoch count. Select the lowest and highest LR-tuned mean terminal validation loss among these six options (3 points each):

Choice	Recipe	Epochs
A	Constant	32
B	Constant	256
C	Constant with EMA	32
D	Constant with EMA	256
E	Linear	32
F	Linear	256

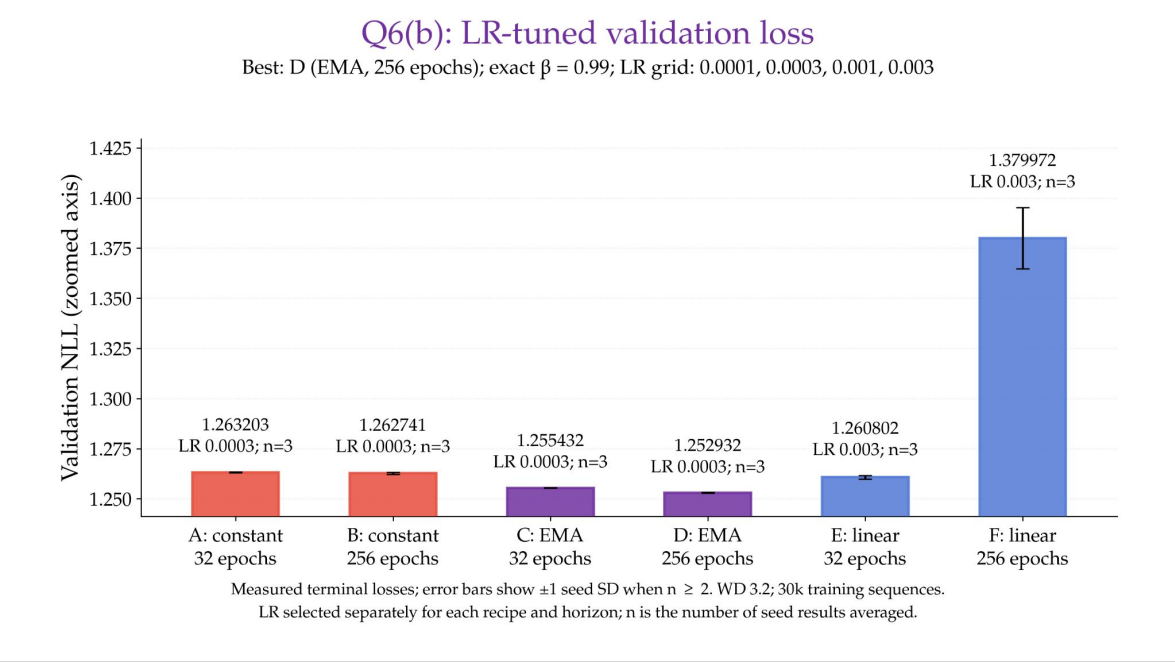
Any option within 0.001 of the corresponding extreme earns full credit. After these full-credit ties, the second-best or second-worst loss tier earns 1.5 points, respectively.

Best loss: A ☐ B ☐ C ☐ D ☐ E ☐ F ☐

Worst loss: A ☐ B ☐ C ☐ D ☐ E ☐ F ☐

Problem 6 - EMA

Answer: Worst is linear with 256 epochs (F), best is EMA with 256 epochs (D)

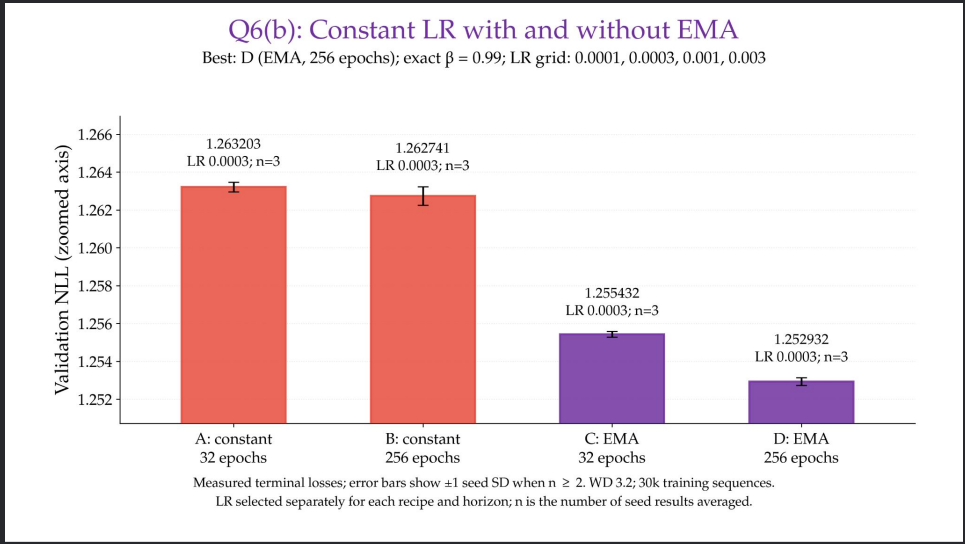


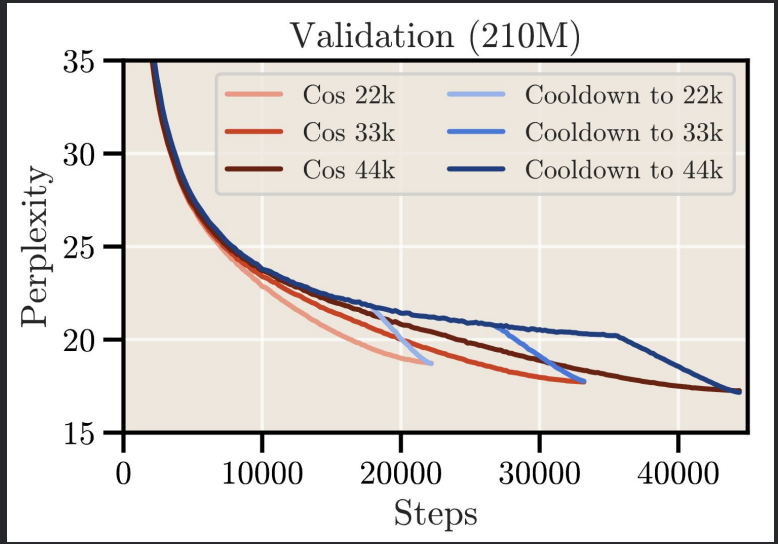
Problem 6 - EMA

Reasoning: We know from earlier that linear learning rate schedule overfits when training for too long

Problem 6 - EMA

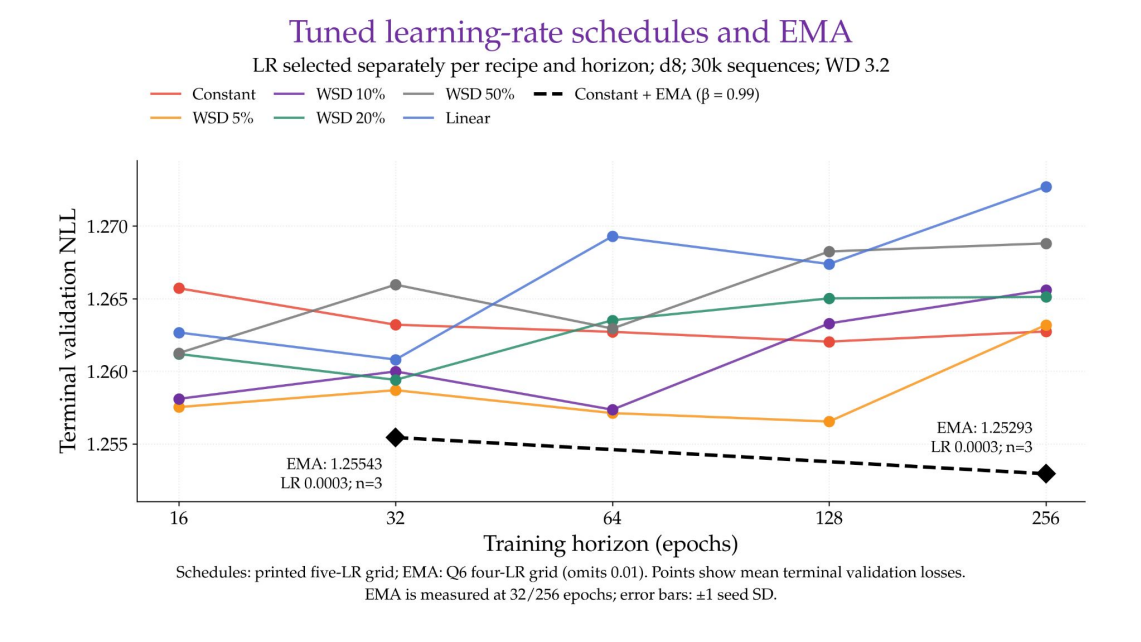
Reasoning: EMA maintains flatness while also getting to the basin and does much better. Interestingly, its loss improves a lot more with epoch count than constant (similar to how LR decay generally has better slope than constant LR)





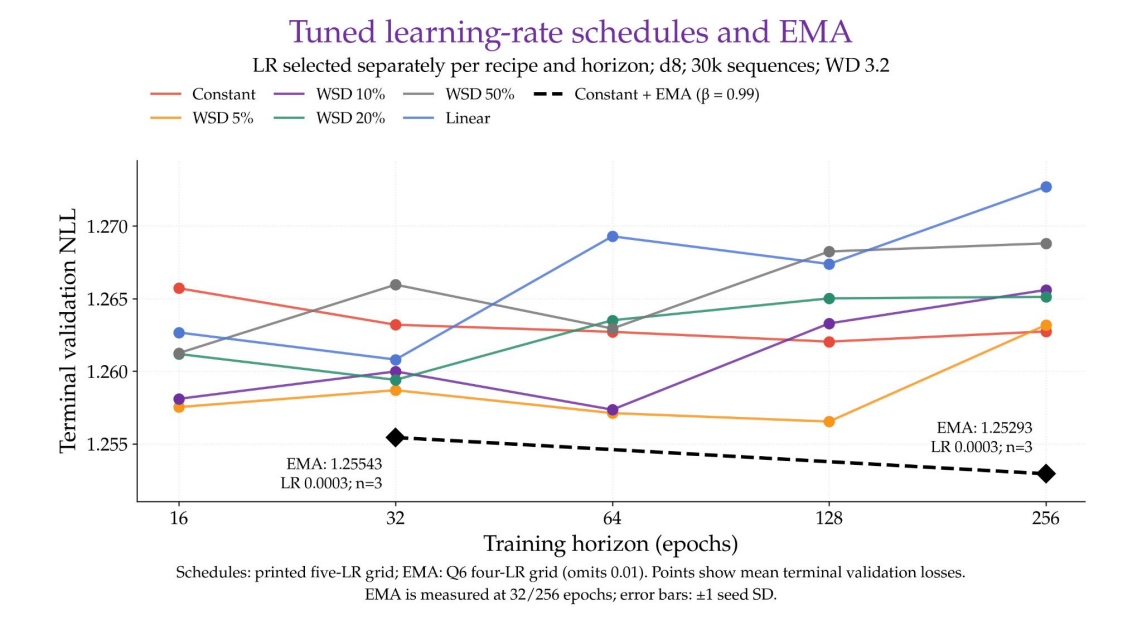
Problem 6 - EMA

Reasoning: EMA seems a lot more effective at handling data-efficient regime relative to all the schedules we tried in earlier question [Izmailov et al, 2018]



Problem 6 - EMA

Leads us into our final question…



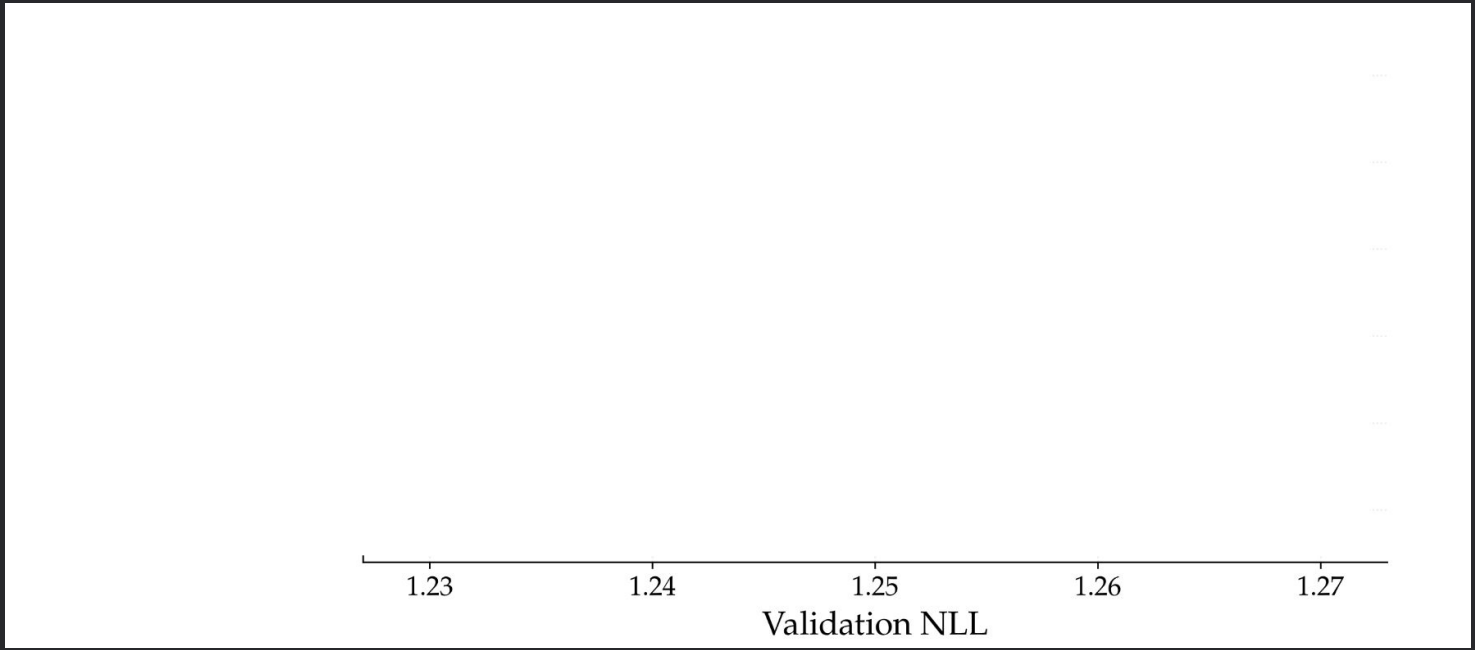
Problem 7 - Everything

7: What is the mega ranking?

Package	Intervention
Q1	Best MoE expert count: the best choice from Q1(a).
Q2	Attention dropout with probability 0.05.
Q3	Related-DNA pretrained initialization, then full fine-tuning on all 30k target sequences for 16 passes with tuned LR.
Q4	Sweep LR, epoch count, and LR schedule: the best choice across Q4(b).
Q5	The best LR-tuned optimizer from Q5(b).
Q6	Sweep LR for constant, linear, and constant with EMA at 32 or 256 epochs: the best choice from Q6(b).

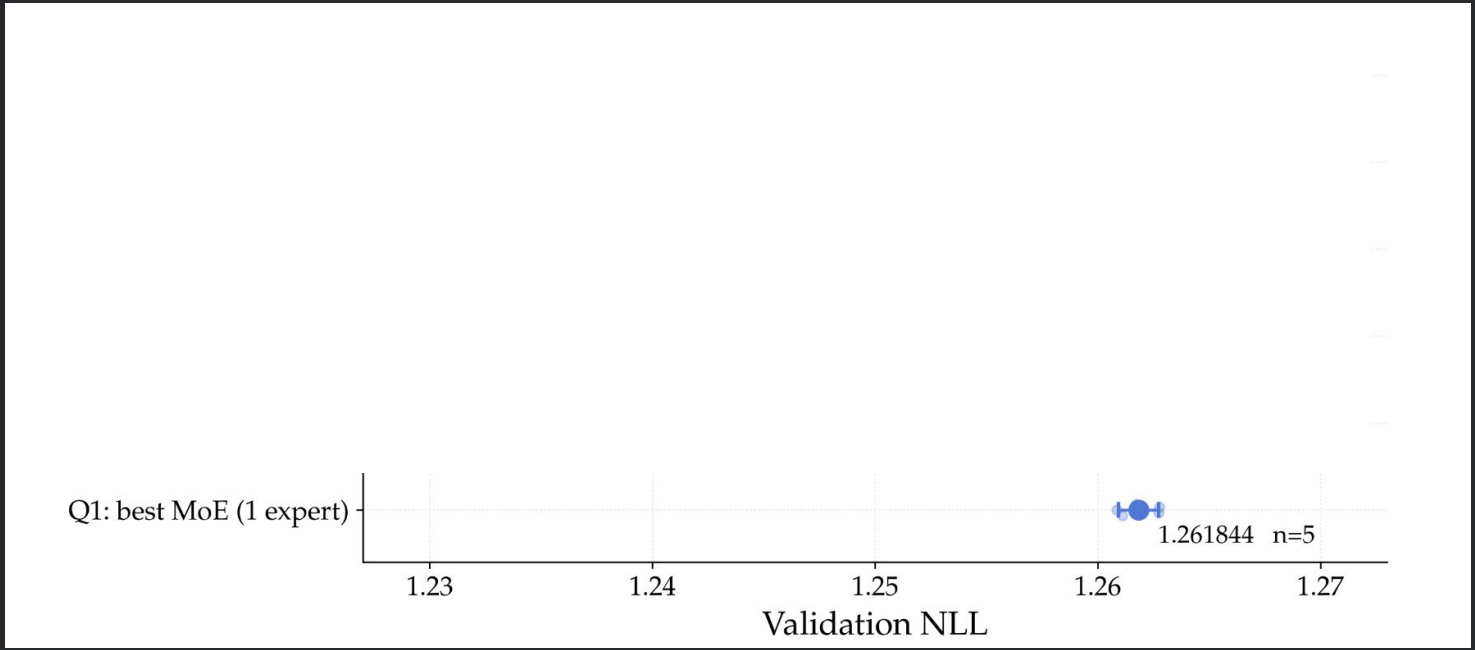
Problem 7 - Everything

Answer: We'll reveal one position at a time



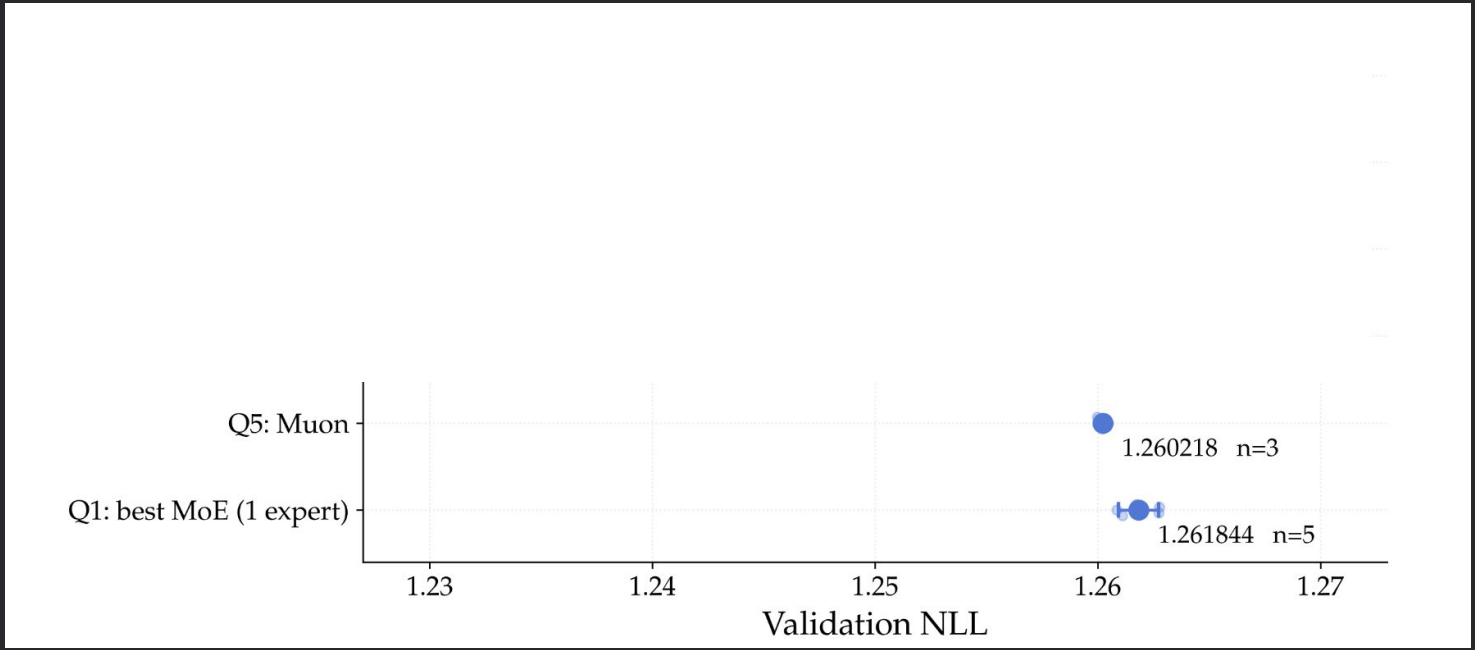
Problem 7 - Everything

6th place: Q1. Best MoE expert count is 1, so this option corresponds to our AdamW baseline



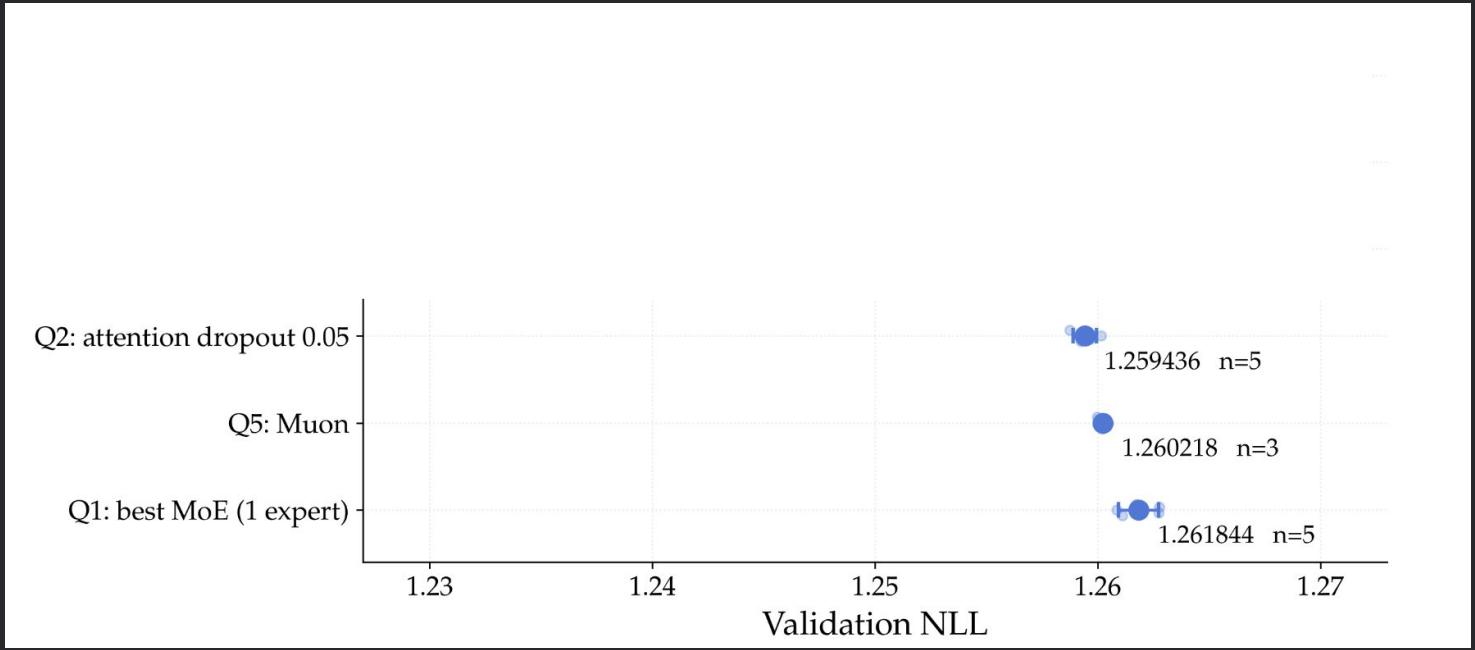
Problem 7 - Everything

5th place: Q5. Muon is an improvement over our standard optimizer in this setting



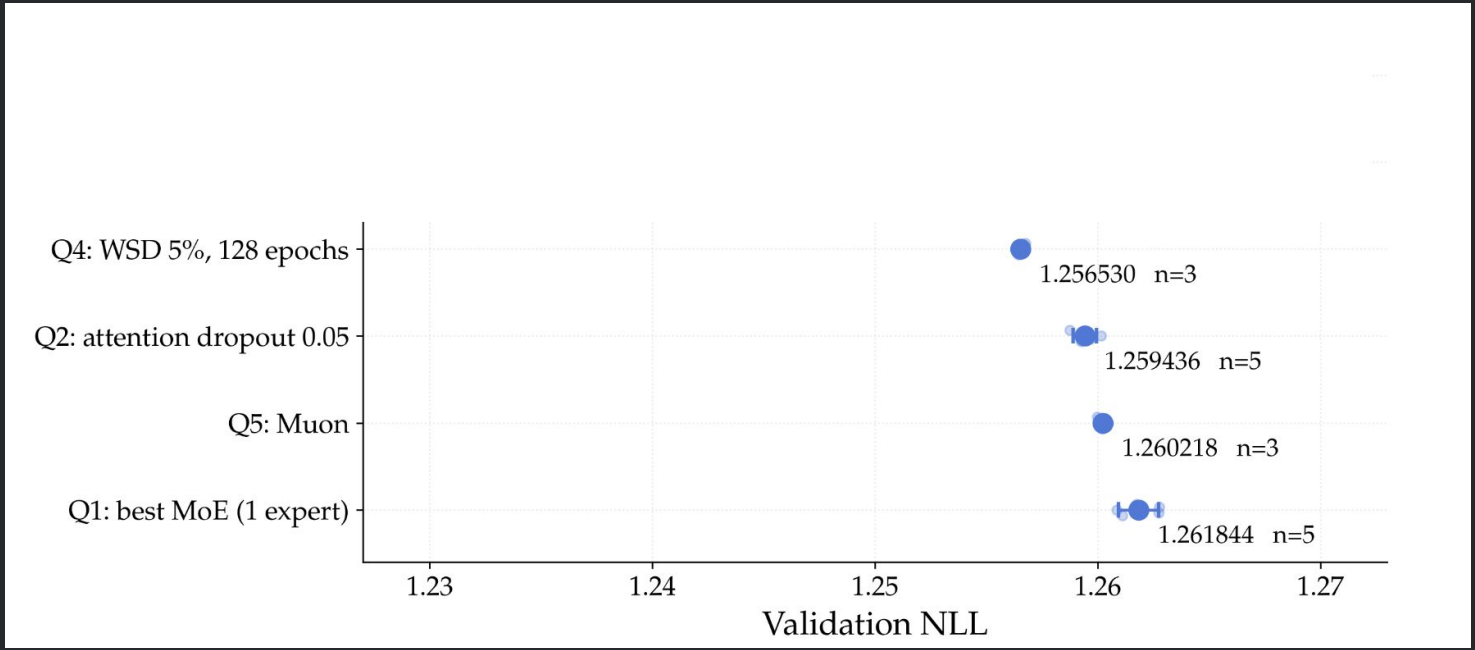
Problem 7 - Everything

4th place: Q2. The regularization from attention dropout designed for data-constrained regime improves loss more than Muon



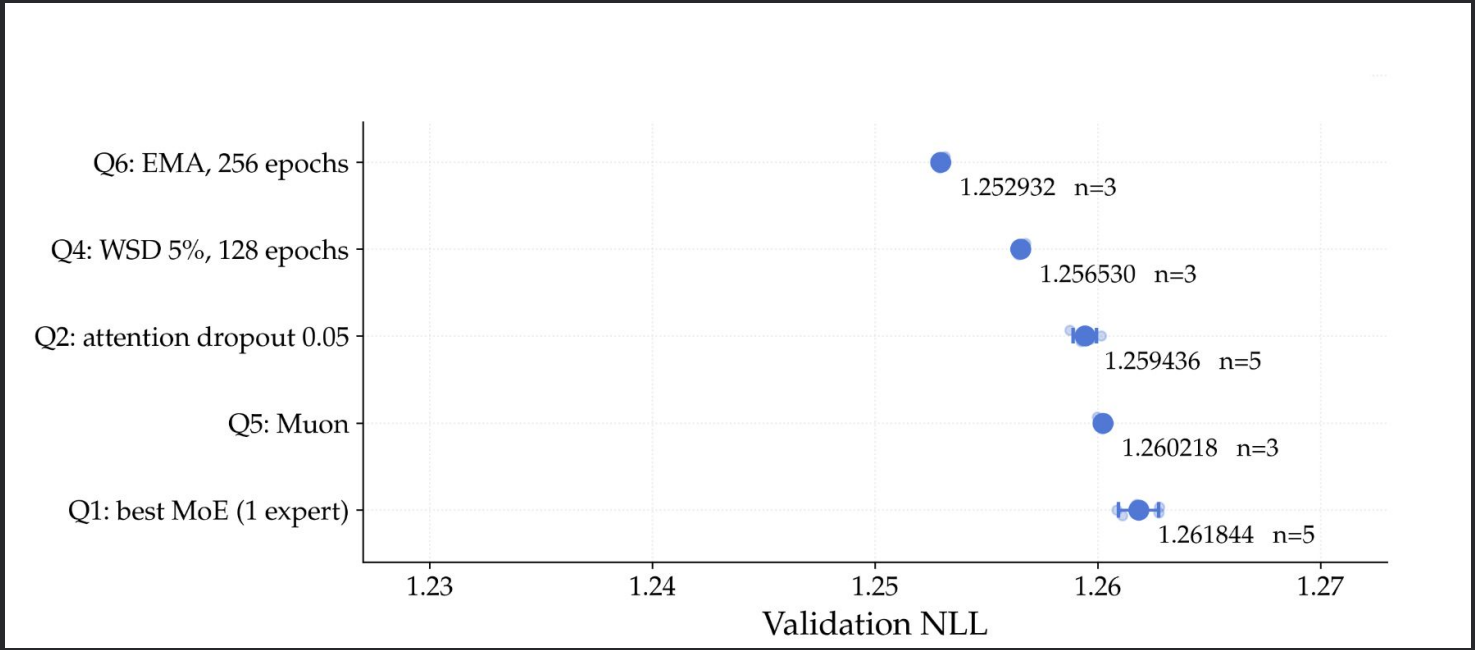
Problem 7 - Everything

3rd place: Q4. Tuning LR and WSD schedule provide more important sharpness regularization relative to dropout



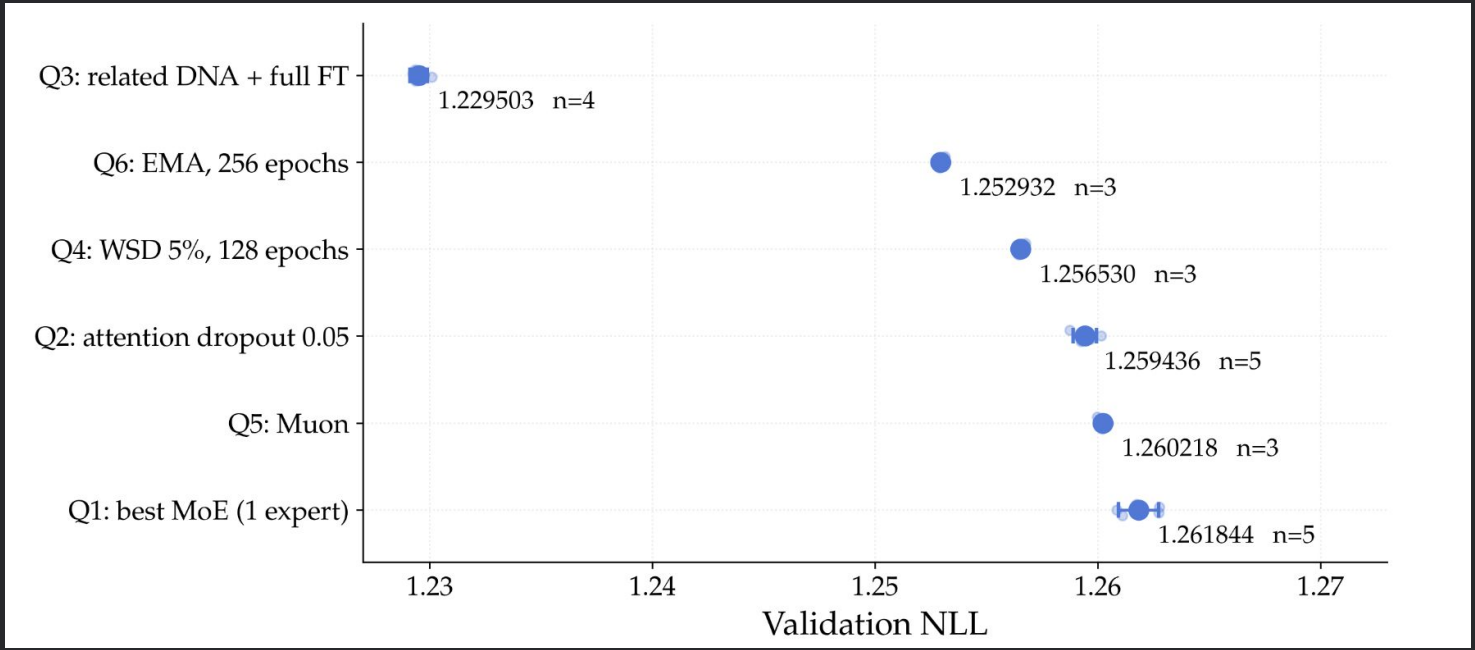
Problem 7 - Everything

2nd place: Q6. As seen in Q6, EMA is a more effective way of scheduling optimization relative to WSD



Problem 7 - Everything

1st place: Q3. Pre-training remains supreme.



Next week

We will be covering something that looks *very* different from pre-training: reinforcement learning

We will explore RL in the context of control tasks

Quiz will be 12/03

Have a good thanksgiving break, and thanks for being our students!

