Outcome Prediction Quiz: Optimization 2

CS 312 · 130 points

Answer sheet. Write your Name and SUNet ID, mark every answer and write your rationale on the separate answer sheet at the end of this booklet (pages S1–S2). **Only the answer sheet is graded.**

Instructions. Predict the outcomes of the following experiments. Each question will have 3 instructions

- **From the assignment** provides some context, providing any needed information from the assignment.
- **This question changes** provides a natural language description of what is different from the assignment.
- **The diff** is the exact code that we run when creating the solution key (printed in Appendix A). It is a patch to the released A3 starter (`a3_release` at `57c60bb`).

Check one box in every row of the answer sheet. Options that are given as sketches are judged on matching shape; their axis values show rough magnitudes. **Scoring:** In rows whose options are ordinal choices, the adjacent option will receive half the maximal points. Each question has three rationale fields: the assignment problem you cite, what you observed in that experiment, and your rationale for choosing your answer based on those observations. Detailed proper citation, with experimental details and sound rationale can return 75% of missed points on a question.

Default setup. The standard setup of the Optimization 2 assignment: Llama-style architecture with 8 layers, hidden size 512, 8 attention heads, QK-norm, a 4k vocab size tokenizer, and about 35M parameters (`d8`); 614.4M training tokens (9,375 updates) of fresh data; AdamW with $\beta_1, \beta_2 = (0.9, 0.95)$ and $\epsilon = 10^{-8}$, linear decay learning rate schedule with 1% linear warmup (93 updates), peak learning rate .003, context length 1024, batch size 64, weight decay 0.1 (masked: hidden and readout matrices decay; embeddings and normalization gains do not), and 1.0 grad clipping; model and data seed 42. Lower validation loss is better. Seed-to-seed noise in the final validation loss is about .003.

Conventions. We follow the same conventions as the rest of A3.

- `wsd`X **schedule**: after the warmup, the peak LR is held constant and then decays linearly to zero over the final fraction X of training (`wsd0.2` decays from update 7,500).
- **Regret** (Problem 2(c)): a run's validation loss minus the control run's at the same update.
- **Sharpness**: $\lambda_{\max}$ is the top eigenvalue of the loss Hessian, measured in FP32 on the 64-sequence held-out probe. A **family block** is the Hessian restricted to one parameter family (embedding, attention, MLP, RMSNorm gains, LM head), as in Problem 4(a).
- **Full-batch runs** (Problem 3(a)): plain gradient descent on the first N training sequences, embedding initialized 16× larger, no clipping, no weight decay, constant LR η.

Prediction

Q1. Basins

Difficulty: Easy

20 points

From the assignment (reference only). Recall the forked-run sweep of Problem 1: the default recipe trains a LM at a constant LR of .003 after the warmup. A shared prefix trains for x updates from model seed 42, then two branches continue for y more updates with different data orders (`batch_order_seed` 1 and 2). For each cell we compare three validation losses: *Best*, the better of the two final losses; *Ens*, the 0.5/0.5 prediction ensemble of the two endpoints; and *Avg*, the 0.5/0.5 weight average.

The diff behind this question is Appendix A.1, page 11.

(a) Average, ensemble, or neither, at a higher LR (12 points, 4 per cell).

This question changes. The whole sweep (prefix and both branches) is repeated at a constant LR of **.01** instead of .003, with the same warmup, model seed and data orders.

At LR .003 (Problem 1's grid) the three cells below came out as shown. For each cell, which ordering, lowest (best) first, do you predict at LR .01? Two losses count as equal (=) when within .01.

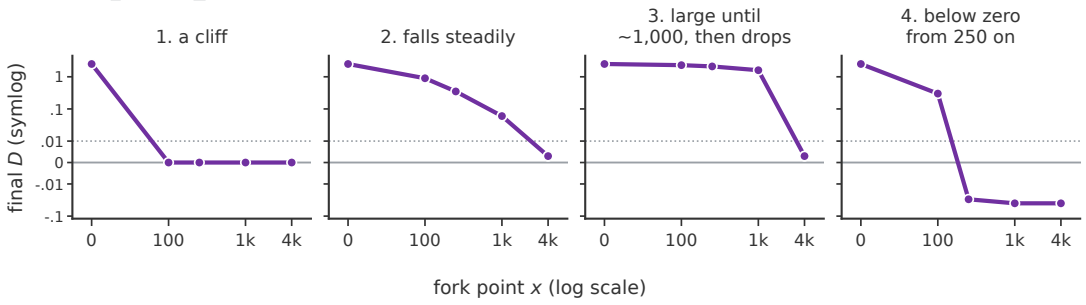
Cell	x	y	At LR .003
(i)	100	50	Avg = Ens < Best
(ii)	100	250	Ens < Avg < Best
(iii)	250	250	Ens < Avg < Best

· 1. Avg < Ens < Best · 4. Ens < Best < Avg
· 2. Avg = Ens < Best · 5. Ens < Avg = Best
· 3. Ens < Avg < Best

(b) How much shared training is enough? (8 points).

This question changes. The **full default recipe** (peak LR .003, linear decay to zero, 9,375 updates) is trained with `batch_order_seed` 1 from model seed 42. Second runs **fork from its checkpoint** (weights and optimizer state) **at update** $x \in \{0, 100, 250, 1{,}000, 4{,}000\}$ ($x = 0$: a fresh run from the same init) and train the remaining $y = 9{,}375 - x$ updates with `batch_order_seed` 2. All runs end at a validation loss of about 2.93.

Let $D(x) = L(\text{Avg}) - \min(L_1, L_2)$ for the two final models. Which sketch of D against the fork point x (log axis) matches? The D axis is log scale, linear within $\pm.01$; the dotted line marks $D = .01$.



· 1. · 2. · 3. · 4.

Mark your answers and write your rationale on answer sheet page S1.

Q2. Fixing a mis-set peak LR mid-run

Difficulty: Hard

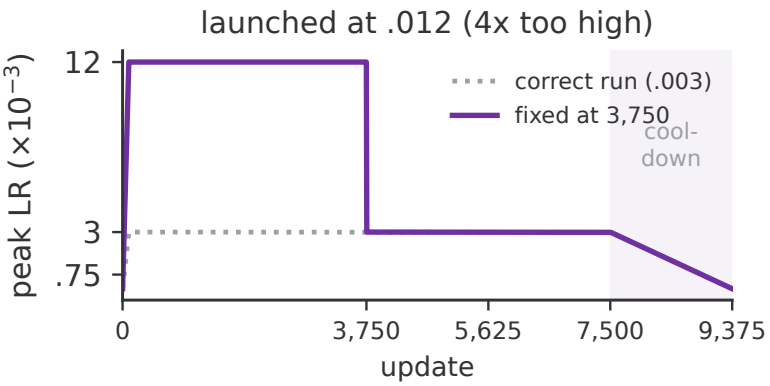
20 points

From the assignment (reference only). Recall the schedule experiments of Problem 2. With the `wsd0.2` schedule at peak LR .003, the final validation loss is 2.935. In the dip-and-rewarm experiments of Problem 2(c), we lowered this run's LR for part of training and brought it back to .003 to see the impact of LR changes.

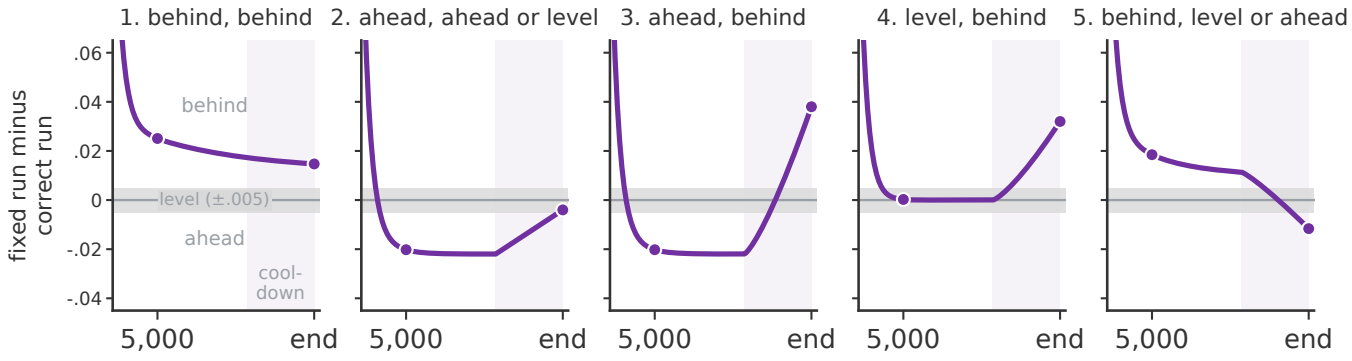
The diff behind this question is Appendix A.2, page 12.

(a) A too-high start, before and after the cooldown (10 points).

This question changes. The `wsd0.2` run is launched with the **wrong peak LR**, .012 (4× too high). Partway through, the mistake is caught and the LR is **set to** .003 for the rest of training, with the same cooldown.



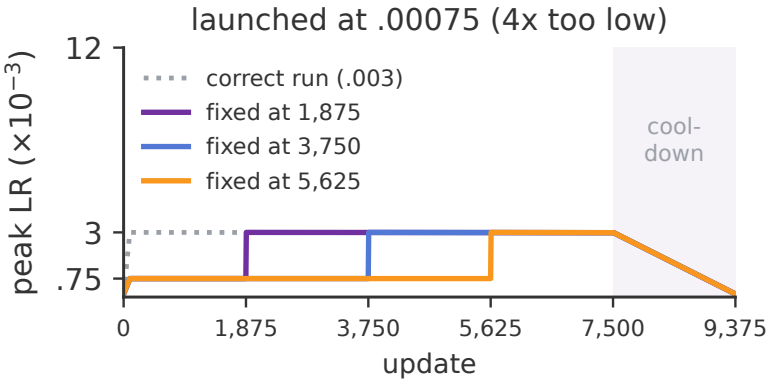
The LR .012 run is fixed at update 3,750 (See Figure). Compare its validation loss with the baseline run (`wsd0.2` with LR .003) at update 5,000 (before the cooldown) and at the end of training. The fixed run is *ahead* if its loss is lower by more than .005, *level* if the two are within .005, and *behind* otherwise. Which sketch matches the fixed run at update 5,000 and at the end?



· 1. · 2. · 3. · 4. · 5.

(b) A too-low start, fixed early or late (10 points).

This question changes. The `wsd0.2` run is launched at .00075 (4× too low) and fixed the same way, at one of three points, or left alone.



Consider the four 0.00075 LR runs: three are fixed at updates 1,875, 3,750 or 5,625 (20%, 40%, 60% of training, see figure) and one is left alone. Rank the choices of when (and whether) to fix the LR based on validation loss. Each option has the lowest loss choice first.

- 1. 20% < 40% < 60% < left alone
- 2. 20% < 40% < left alone < 60%
- 3. 20% < left alone < 40% < 60%
- 4. 20% = 40% = 60% (within .005) < left alone

Mark your answers and write your rationale on answer sheet page S1.

DO NOT ANSWER

Q3. Weight averaging with an EMA

Difficulty: Medium

20 points

From the assignment (reference only). Recall the evaluation-only EMA of the weights in Problem 2(b) (decay .999, a window of about 1,000 updates). It helps a lot at constant LR and slightly hurts at the end of schedules that decay to zero. At a constant LR of .003 after the warmup, the final validation loss is 3.099 for the raw weights and 2.973 for the EMA.

The diff behind this question is Appendix A.3, page 12.

(a) The best peak LR for constant LR plus EMA (10 points).

This question changes. The constant-LR run at peak LR **.0015 and .006**, with the same EMA.

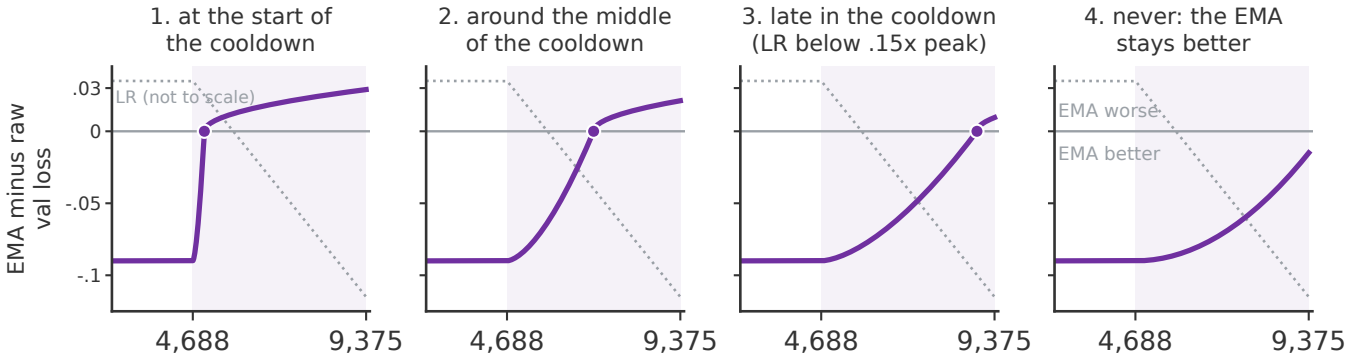
Let $L_{.0015}$, $L_{.003}$ and $L_{.006}$ be the final validation losses of the EMA weights at the three peak LRs. Which ranking do you predict?

· 1. $L_{.0015} < L_{.003} < L_{.006}$ · 2. $L_{.003}$ is the lowest of the three
· 3. $L_{.006} < L_{.003} < L_{.0015}$ · 4. All three within .005

(b) When the EMA stops helping (10 points).

This question changes. The `wsd0.5` schedule at peak LR .003 (constant, then a linear decay to zero over the last 50% of training, from update 4,688), with the same EMA.

In the `wsd0.5` run, let $\Delta(t)$ be the EMA's validation loss minus the raw weights' at update t. When does Δ become positive (EMA worse) for good?



· 1. · 2. · 3. · 4.

Mark your answers and write your rationale on answer sheet page S1.

DO NOT WRITE HERE

Q4. Dips and rewarms

Difficulty: Medium

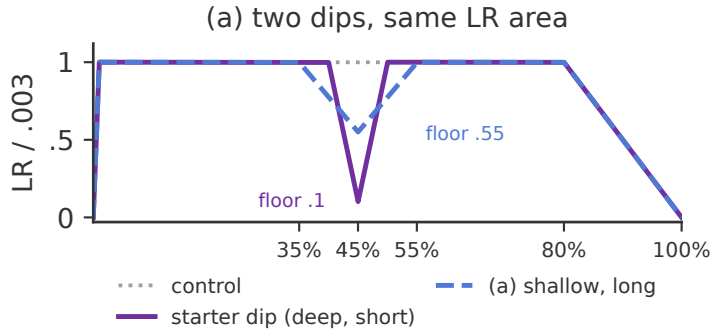
20 points

From the assignment (reference only). Recall the dip-and-rewarm experiments of Problem 2(c). A dip multiplies the LR of the `wsd0.2` control (peak .003) by a factor that falls linearly from 1 to a floor and then returns linearly to 1. The starter dip runs from 40% to 50% of training with a floor of 0.1× at 45%, and gives a large temporary gain at its bottom.

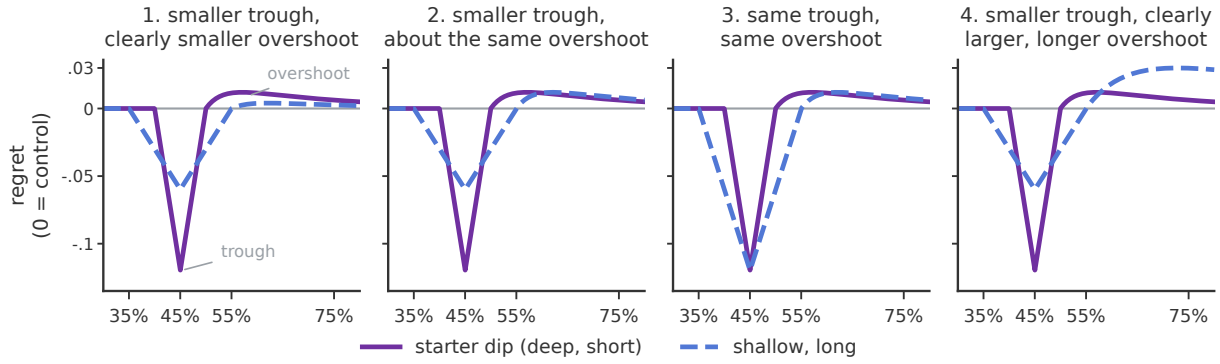
The diff behind this question is Appendix A.4, page 13.

(a) Same area, different shape (10 points).

This question changes. A **shallow, long dip** from 35% to 55% of training with a floor of 0.55× at 45%. It removes the same LR area as the starter dip (422 updates' worth of peak LR).



Which sketch matches the two dips' regret curves? ("About the same overshoot": within a factor of 1.5.)

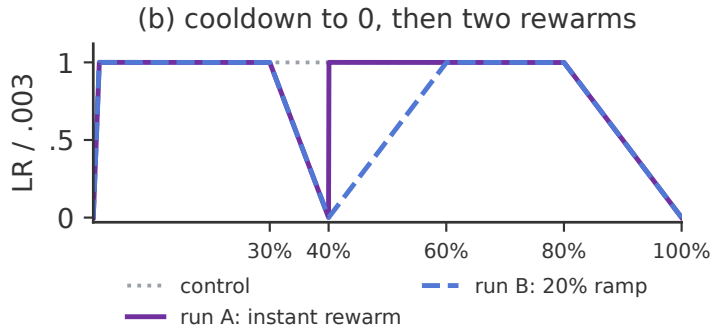


· 1. · 2. · 3. · 4.

DO NOT SHARE

(b) Instant or gradual rewarm (10 points).

This question changes. Two runs on the control: run A cools **linearly to zero** between 30% and 40% of training and then **jumps back** to .003; run B cools the same way and then **re-warms linearly** from 0 to .003 between 40% and 60%.



Let L_A and L_B be the final validation losses of runs A and B. Which interval contains $L_A - L_B$?

· 1. $(-\infty, -.005)$ · 2. $[-.005, .005)$ · 3. $[.005, \infty)$

Mark your answers and write your rationale on answer sheet page S2.

DO NOT ANSWER

Q5. Full-batch gradient descent at the edge

Difficulty: Medium

20 points

From the assignment (reference only). Recall the full-batch experiments of Problem 3(a): gradient descent on the first N training sequences for 1,500 updates, with $\lambda_{\max}$ measured on the training set every 25 updates. In the clean case ($N = 32$, $\eta = .01$, `d8`), $\eta\lambda_{\max}$ first exceeds 2 at update 375.

The diff behind this question is Appendix A.5, page 13.

(a) How far above 2 at a large step size (10 points).

This question changes. $N = \mathbf{256}$ sequences and $\eta = \mathbf{0.2}$.

From update 100 on, while the training loss is still above 1 (before the sequences are memorized), which interval contains the median of $\eta\lambda_{\max}$?

· 1. $(0, 2.2)$ · 2. $[2.2, 3)$ · 3. $[3, 6)$ · 4. $[6, \infty)$ · 5. The run diverges

(b) Depth and progressive sharpening (10 points).

This question changes. The clean case at **depth 4 and depth 16** (width 512, 64-dimensional heads), next to the standard depth 8.

Let t_L be the first update at which $\eta\lambda_{\max} > 2$ at depth L. Which ranking do you predict?

· 1. $t_4 < t_8 < t_{16}$ · 2. All three within 20% of each other
· 3. $t_{16} < t_8 < t_4$ · 4. t_8 is the earliest of the three

Mark your answers and write your rationale on answer sheet page S2.

DO NOT ANSWER HERE

Q6. Weight decay as a sharpness knob

Difficulty: Medium

20 points

From the assignment (reference only). Recall Problem 4(b): weight decay moves $\lambda_{\max}$ at nearly fixed loss through rescaling symmetries. The default mask decays the hidden and readout matrices (including the LM head) but not the embedding or the RMSNorm gains. With this mask and peak LR .003, weight decay 0, .1 and .3 give a final $\lambda_{\max}$ of about 2.5, 6 and 15.

The diff behind this question is Appendix A.6, page 13.

(a) Decaying every tensor (10 points).

This question changes. Weight decay .1 is applied to **every tensor**, including the embedding and all RMSNorm gains.

Which interval contains the run's final $\lambda_{\max}$ divided by the default run's (same seed)?

· 1. $(0, .8)$ · 2. $[.8, 1.2)$ · 3. $[1.2, \infty)$

(b) Trading LR for weight decay (10 points).

This question changes. The default mask, with the peak LR η and the weight decay w changed **together at a fixed product** $\eta w = 3 \times 10^{-4}$, the default value:

Run	Peak LR η	Weight decay w	ηw
S	.001	.3	3×10^{-4}
Default	.003	.1	3×10^{-4}
T	.01	.03	3×10^{-4}

Let λ_S, λ_{def} and λ_T be the final $\lambda_{\max}$ of runs S, Default and T. Which ranking do you predict?

· 1. All three within 20% of each other · 2. $\lambda_S > \lambda_{\text{def}} > \lambda_T$
· 3. $\lambda_T > \lambda_{\text{def}} > \lambda_S$ · 4. λ_{def} is the highest of the three

Mark your answers and write your rationale on answer sheet page S2.

DO NOT ANSWER HERE

Q7. Training in the top subspace

Difficulty: Hard

10 points

From the assignment (reference only). Recall Problem 4(c). From the SGDM checkpoint at 100M tokens, removing the component of each update in the top-10 Hessian eigenspace costs nothing over 300 steps, while keeping only that component makes no progress.

The diff behind this question is Appendix A.7, page 14.

Adam fed only the top subspace (10 points).

This question changes. The keep-only experiment on the **AdamW checkpoint** at 100M tokens: the minibatch gradient is replaced by its component in the top-10 eigenspace of the Hessian **before AdamW sees it**, for 300 steps, with the eigenvectors recomputed every 25 steps. AdamW's moment estimates are carried over from the checkpoint, not reset.

Which interval contains the validation loss after the 300 steps minus the validation loss at the checkpoint (positive if the loss went up)?

· 1. $(-\infty, -.05)$ · 2. $[-.05, .05)$ · 3. $[.05, .3)$ · 4. $[.3, \infty)$

Mark your answers and write your rationale on answer sheet page S2.

DO NOT ANSWER HERE

Appendix: the diffs behind the questions

Each question's runs are exactly these patches to the released A3 starter (the `assignments` repository, branch `a3_release` at `57c60bb`); each applies with `git apply`. Red lines are removed, green lines are added. The answer key comes from these runs.

A.1 — Q1: Basins

Part (a):

```
--- a/experiments/a3_optimization/p1_mode_connectivity.py
+++ b/experiments/a3_optimization/p1_mode_connectivity.py
@@ -62,2 +62,11 @@ BRANCHES = {

+# Quiz 3, Question 1(a): the same sweep at a constant LR of .01 (a Problem 1(d) knob), measured on
+# the cells (100, 50), (100, 250) and (250, 250).
+LR01 = replace(BASE, learning_rate=0.01)
+PREFIX_LR01 = prefix_for(LR01)
+BRANCHES_LR01 = {x: [branch(x, seed, base=LR01) for seed in BRANCH_ORDER_SEEDS] for x in (100, 250)}
+LR01_RUNS = [PREFIX_LR01] + [run for runs in BRANCHES_LR01.values() for run in runs]
+LR01_PAIRS = [((training_run_name(a), x + y), (training_run_name(b), x + y))
+              for x, y in ((100, 50), (100, 250), (250, 250)) for a, b in [BRANCHES_LR01[x]]]
+

@@ -107,3 +116,3 @@ def main():

-        launch_training_jobs([PREFIX])
+        launch_training_jobs([PREFIX, PREFIX_LR01])
     elif args.stage == "branches":
@@ -112,3 +121,4 @@ def main():
         check_prefix_checkpoints()
-        launch_training_jobs([run for runs in BRANCHES.values() for run in runs])
+        check_prefix_checkpoints(PREFIX_LR01)
+        launch_training_jobs([run for runs in BRANCHES.values() for run in runs] + LR01_RUNS[1:])
     else:
@@ -116,3 +126,3 @@ def main():

-        rows = measure_pairs_on_modal(pairs())
+        rows = measure_pairs_on_modal(pairs() + LR01_PAIRS)
         RESULTS_PATH.parent.mkdir(parents=True, exist_ok=True)
```

Part (b):

```
--- a/experiments/a3_optimization/p1_mode_connectivity.py
+++ b/experiments/a3_optimization/p1_mode_connectivity.py
@@ -62,2 +62,11 @@ BRANCHES = {

+# Quiz 3, Question 1(b): the default recipe in data order 1, and a second run in data order 2
+# forked from it at update x (x = 0: a fresh run from the same initialization).
+FORKS = (0, 100, 250, 1000, 4000)
+FULL = TrainConfig(batch_order_seed=1, keep_checkpoint_steps=FORKS[1:], run_name_suffix="quiz3-q1b")
+SECOND = [replace(FULL, batch_order_seed=2, keep_checkpoint_steps=(),
+                  **(dict(fork_from_run=training_run_name(FULL), fork_from_step=x) if x else {}))
+          for x in FORKS]
+FINAL_PAIRS = [((training_run_name(FULL), None), (training_run_name(r), None)) for r in SECOND]
+

@@ -107,3 +116,3 @@ def main():

-        launch_training_jobs([PREFIX])
+        launch_training_jobs([PREFIX, FULL])
     elif args.stage == "branches":
```

DO NOT ANSWER HERE

```
@@ -112,3 +121,3 @@ def main():
         check_prefix_checkpoints()
-        launch_training_jobs([run for runs in BRANCHES.values() for run in runs])
+        launch_training_jobs([run for runs in BRANCHES.values() for run in runs] + SECOND)
     else:
@@ -116,3 +125,3 @@ def main():

-        rows = measure_pairs_on_modal(pairs())
+        rows = measure_pairs_on_modal(pairs() + FINAL_PAIRS)
         RESULTS_PATH.parent.mkdir(parents=True, exist_ok=True)
```

A.2 — Q2: Fixing a mis-set peak LR mid-run

```
--- a/experiments/a3_optimization/p2_river_valley.py
+++ b/experiments/a3_optimization/p2_river_valley.py
@@ -5,4 +5,6 @@

+import sys
+
 from modal_train import launch_training_jobs
-from train import TrainConfig
+from train import TrainConfig, training_run_name

@@ -40,2 +42,11 @@ RUNS = SCHEDULE_RUNS + REWARM_RUNS + HORIZON_FREE_RUNS

+# Quiz 3, Question 2: wsd0.2 launched at a wrong peak LR (4x too high or too low). The fix forks
+# the run at update u with the peak LR set to .003 (stage 'quiz3-q2', after these runs finish).
+WRONG = [TrainConfig(lr_schedule="wsd0.2", learning_rate=lr, keep_checkpoint_steps=u,
+                     run_name_suffix="quiz3-q2")
+         for lr, u in ((0.012, (3750,)), (0.00075, (1875, 3750, 5625)))]
+FIXED = [TrainConfig(lr_schedule="wsd0.2", fork_from_run=training_run_name(run), fork_from_step=u,
+                     run_name_suffix="quiz3-q2") for run in WRONG for u in run.keep_checkpoint_steps]
+RUNS += WRONG
+

@@ -49,3 +60,3 @@ RUNS = SCHEDULE_RUNS + REWARM_RUNS + HORIZON_FREE_RUNS
 def main():
-    launch_training_jobs(RUNS)
+    launch_training_jobs(FIXED if sys.argv[1:] == ["quiz3-q2"] else RUNS)
```

A.3 — Q3: Weight averaging with an EMA

```
--- a/experiments/a3_optimization/p2_river_valley.py
+++ b/experiments/a3_optimization/p2_river_valley.py
@@ -40,2 +40,7 @@ RUNS = SCHEDULE_RUNS + REWARM_RUNS + HORIZON_FREE_RUNS

+# Quiz 3, Question 3: constant LR at .0015 and .006, and wsd0.5 at .003, with the same EMAs.
+RUNS += [TrainConfig(lr_schedule=schedule, learning_rate=lr, ema_decays=EMA_DECAYS,
+                     run_name_suffix="quiz3-q3")
+         for schedule, lr in (("wsd0.0", 1.5e-3), ("wsd0.0", 6e-3), ("wsd0.5", 3e-3))]
+
```

DO NOT ANSWER HERE

A.4 — Q4: Dips and rewarms

```
--- a/experiments/a3_optimization/p2_river_valley.py
+++ b/experiments/a3_optimization/p2_river_valley.py
@@ -40,2 +40,7 @@ RUNS = SCHEDULE_RUNS + REWARM_RUNS + HORIZON_FREE_RUNS

+# Quiz 3, Question 4: (a) a shallow, long dip with the part (c) dip's LR area; (b) a cooldown to 0
+# over 30-40% of training, then an instant (A) or a 40-60% linear (B) return to the peak LR.
+RUNS += [TrainConfig(lr_schedule="wsd0.2", lr_dip=dip, run_name_suffix="quiz3-q4")
+         for dip in ((0.35, 0.45, 0.55, 0.55), (0.3, 0.4, 0.4, 0.0), (0.3, 0.4, 0.6, 0.0))]
+
```

A.5 — Q5: Full-batch gradient descent at the edge

```
--- a/experiments/a3_optimization/p3_edge_of_stability.py
+++ b/experiments/a3_optimization/p3_edge_of_stability.py
@@ -8,4 +8,7 @@ logging/sharpness/*. Read the results back with experiments.a3_optimization.shar

+from dataclasses import replace
+
 from metric_logging import AFTER_BACKWARD, MetricLogger
 from modal_train import launch_training_jobs
+from model_config import depth_model_config
 from sharpness_logging import SharpnessLogger
@@ -79,2 +82,8 @@ FULL_BATCH_RUNS = [
     full_batch(32, 1500, learning_rate=0.01),  # (a) i: the clean case
+    # Quiz 3, Question 5: (a) N = 256 at eta = 0.2; (b) the clean case at depth 4 and 16.
+    full_batch(256, 1500, learning_rate=0.2, run_name_suffix="quiz3-q5a"),
+    *(full_batch(32, 1500, learning_rate=0.01, run_name_suffix="quiz3-q5b",
+                 model_config=replace(depth_model_config(8), name=f"d{depth}w512",
+                                      num_hidden_layers=depth))  # width 512 at every depth
+      for depth in (4, 16)),
 ]
```

A.6 — Q6: Weight decay as a sharpness knob

Part (a):

```
--- a/experiments/a3_optimization/p4_inside_the_hessian.py
+++ b/experiments/a3_optimization/p4_inside_the_hessian.py
@@ -56,2 +56,3 @@ RUNS = [
     # TrainConfig(<setting>=<value>, wandb_tags=(EXPERIMENT_KEY,)),
+    TrainConfig(run_name_suffix="quiz3-q6a"),  # Quiz 3, Question 6(a)
 ]
--- a/optimizers.py
+++ b/optimizers.py
@@ -13,2 +13,3 @@ def is_norm_module(module):
 def should_apply_weight_decay(module, parameter_name):
+    return True  # Quiz 3, Question 6(a): decay every tensor
     if isinstance(module, torch.nn.Embedding):
```

Part (b):

```
--- a/experiments/a3_optimization/p4_inside_the_hessian.py
+++ b/experiments/a3_optimization/p4_inside_the_hessian.py
@@ -56,2 +56,5 @@ RUNS = [
     # TrainConfig(<setting>=<value>, wandb_tags=(EXPERIMENT_KEY,)),
+    # Quiz 3, Question 6(b): peak LR and weight decay at the default product, 3e-4.
+    TrainConfig(learning_rate=1e-3, weight_decay=0.3, run_name_suffix="quiz3-q6b"),
+    TrainConfig(learning_rate=1e-2, weight_decay=0.03, run_name_suffix="quiz3-q6b"),
 ]
```

DO NOT ANSWER HERE

A.7 — Q7: Training in the top subspace

```
--- a/experiments/a3_optimization/p4_inside_the_hessian.py
+++ b/experiments/a3_optimization/p4_inside_the_hessian.py
@@ -74,2 +74,4 @@ SUBSPACE_JOBS = [
 ]
+# Quiz 3, Question 7: the top arm on the AdamW checkpoint.
+SUBSPACE_JOBS += [{"name": "quiz3-q7a", "continue": {**START, "run": "adamw", "transform": "top"}}]
```

DO NOT ANSWER HERE

Name _____ **SUNet ID** _____

your Stanford login, e.g. `jdoe`; not the 8-digit ID number

Check one box in every row and fill in all three rationale fields of each question. Only these two pages are graded.

Q1. Basins — 20 points

(a) at LR .01 4 points per row	1. Avg<Ens<Best	2. Avg=Ens<Best	3. Ens<Avg<Best	4. Ens<Best<Avg	5. Ens<Avg=Best
(i) $x = 100$, $y = 50$	☐	☐	☐	☐	☐
(ii) $x = 100$, $y = 250$	☐	☐	☐	☐	☐
(iii) $x = 250$, $y = 250$	☐	☐	☐	☐	☐

(b) sketch of $D(x)$ 8 points ☐1. a cliff ☐2. falls steadily ☐3. large until ~1,000, then drops ☐4. below zero from 250 on

Assignment problem cited	**Observations from the experiment**	**Your rationale**

Q2. Fixing a mis-set peak LR mid-run — 20 points

(a) at 5,000, at the end 10 points ☐1. behind, behind ☐2. ahead, ahead or level ☐3. ahead, behind ☐4. level, behind ☐5. behind, level or ahead

(b) ranking 10 points ☐1. 20% < 40% < 60% < alone ☐2. 20% < 40% < alone < 60% ☐3. 20% < alone < 40% < 60% ☐4. 20% = 40% = 60% < alone

Assignment problem cited	**Observations from the experiment**	**Your rationale**

Q3. Weight averaging with an EMA — 20 points

(a) ranking 10 points ☐1. $L_{.0015} < L_{.003} < L_{.006}$ ☐2. $L_{.003}$ lowest ☐3. $L_{.006} < L_{.003} < L_{.0015}$ ☐4. All within .005

(b) EMA worse for good 10 points ☐1. at the start of the cooldown ☐2. around its middle ☐3. late (LR below .15× peak) ☐4. never

Assignment problem cited	**Observations from the experiment**	**Your rationale**

Name _____ **SUNet ID** _____

your Stanford login, e.g. `jdoe`; not the 8-digit ID number

Q4. Dips and rewarms — 20 points

(a) sketch 10 points ☐1. smaller trough, smaller overshoot ☐2. smaller trough, similar overshoot ☐3. same trough, same overshoot ☐4. smaller trough, larger, longer overshoot

(b) $L_A - L_B$ 10 points ☐1. $(-\infty, -.005)$ ☐2. $[-.005, .005)$ ☐3. $[.005, \infty)$

Assignment problem cited	Observations from the experiment	Your rationale

Q5. Full-batch gradient descent at the edge — 20 points

(a) median $\eta\lambda_{\max}$ 10 points ☐1. $(0, 2.2)$ ☐2. $[2.2, 3)$ ☐3. $[3, 6)$ ☐4. $[6, \infty)$ ☐5. Diverges

(b) ranking 10 points ☐1. $t_4 < t_8 < t_{16}$ ☐2. Within 20% ☐3. $t_{16} < t_8 < t_4$ ☐4. t_8 earliest

Assignment problem cited	Observations from the experiment	Your rationale

Q6. Weight decay as a sharpness knob — 20 points

(a) $\lambda_{\max}$ ratio 10 points ☐1. $(0, .8)$ ☐2. $[.8, 1.2)$ ☐3. $[1.2, \infty)$

(b) ranking 10 points ☐1. Within 20% ☐2. $\lambda_S > \lambda_{\text{def}} > \lambda_T$ ☐3. $\lambda_T > \lambda_{\text{def}} > \lambda_S$ ☐4. λ_{def} highest

Assignment problem cited	Observations from the experiment	Your rationale

Q7. Training in the top subspace — 10 points

loss change 10 points ☐1. $(-\infty, -.05)$ ☐2. $[-.05, .05)$ ☐3. $[.05, .3)$ ☐4. $[.3, \infty)$

Assignment problem cited	Observations from the experiment	Your rationale