Optimization & Uncertainty in Learning Report Rubric
CS7641: Machine Learning
Fall 2025

Last Updated: 08/20/25

Due Dates

- **10/13/2025** Grading Instructions Email sent to Team.
- **10/14/2025** Internal Team Meeting to overview the Rubric and Grading Cycle.
- **10/15/2025** First Two Calibration Reports on your sheet in Teams graded by EOD.
- **10/17/2025** Calibration discussion with your Head TA by EOW. Recalibrate initial two reports and feedback based on Head TA discussion for group homogeneity.
- **10/21/2025** Roughly 33-50% of Reports Graded. (Progress of 17-25 Reports Graded).
- **10/24/2025** Roughly 50-80% of Reports Graded. (Progress of 25-40 Reports Graded).
- **10/28/2025** Full 100% of Reports Graded by EOD.
- **10/29/2025** Stats run. Any outstanding or missed reports. Grades Posted to students.

Tips

Algorithm Review

This section is a quick reference for graders. It summarizes how each method works, what correct disclosures look like, and the most common failure modes to watch for. Use it to sanity check figures, captions, and accounting before you score a section.

Randomized Optimization on the last layers

Randomized Hill Climbing *Idea.* Start from a candidate, propose a local perturbation, accept only if it improves the objective; optionally restart after stagnation.

$$\theta' = \theta + \eta \odot \xi, \quad \text{accept if } \mathcal{L}_{val}(\theta') < \mathcal{L}_{val}(\theta)$$

Expected disclosures. Restart policy, step size schedule, perturbation distribution and scale, adaptation rule for scale. Exact list of unfrozen layers and total trainable parameters. *Good signs.* Best so far validation loss decreases then plateaus, acceptance rate decays as expected, restarts recorded. *Pitfalls to flag.* Objective evaluated in training mode, unequal function evaluation budgets, step sizes so small that nothing moves, tuning more than approximately fifty thousand parameters.

Simulated Annealing *Idea.* Like hill climbing but sometimes accept worse moves with a probability that decays over time to escape local minima.

$$p(\text{accept}) = \begin{cases} 1, & \Delta \le 0 \\ \exp(-\Delta/T), & \Delta > 0 \end{cases} \quad \text{with } T \downarrow$$

Expected disclosures. Initial temperature, cooling schedule, any step size cooling, perturbation distribution and scale. *Good signs.* Early phase accepts occasional worse candidates, later phase converges. Cooling schedule shown or stated. *Pitfalls to flag.* Temperature collapses too quickly so the method becomes greedy, or never cools and keeps wandering.

Genetic Algorithm *Idea.* Maintain a population of weight vectors, select parents, recombine, mutate, and keep the best. *Expected disclosures.* Population size, selection rule, crossover operator, mutation rate and scale, elitism, representation is real coded, bounds or clipping if used. *Good signs.* Population wide improvement on the progress curve, not only a single lucky individual. Elitism recorded if used. *Pitfalls to flag.* Bitstring representation for continuous weights without justification, mutation scale far too small or far too large, evaluating different layers or a different parameter cap than stated.

Evaluation hygiene for Randomized Optimization

- Use `model.eval()` for every objective: dropout disabled, Batch Normalization uses stored running statistics. No data augmentation on validation.
- Count one function evaluation per full validation loss computation. Caching is fine but does not change counts.
- Only the last one to three layers are unfrozen. The parameter count for Randomized Optimization is reported and is at most approximately fifty thousand.
- No gradient steps interleaved with Randomized Optimization.

Common Randomized Optimization grading checks

Equal function evaluation budgets across Randomized Hill Climbing, Simulated Annealing, and Genetic Algorithm. Same unfrozen layers and same parameter cap across algorithms. Plateau stopping, if used, is identical and counted. Progress curves show best so far validation loss versus function evaluations with readable captions that list operator settings.

Optimizer mechanics on the full network

Let $g_t = \nabla_\theta \mathcal{L}_{train}(\theta_t)$, learning rate α, momentum μ, Adam moments β_1, β_2, and ε a small constant.

Stochastic Gradient Descent without momentum

$$\theta_{t+1} = \theta_t - \alpha\, g_t$$

Watch for. Learning rate must be tuned for this method. Often slower to reach a fixed validation loss threshold.

Stochastic Gradient Descent with momentum

$$v_{t+1} = \mu v_t + g_t, \quad \theta_{t+1} = \theta_t - \alpha\, v_{t+1}$$

Watch for. Momentum value reported. Same initialization and seeds as other methods.

Nesterov momentum

$$v_{t+1} = \mu v_t + \nabla_\theta \mathcal{L}_{train}(\theta_t - \alpha\mu v_t), \quad \theta_{t+1} = \theta_t - \alpha\, v_{t+1}$$

Watch for. Look ahead gradient is used, not plain momentum under a different name.

Adam baseline with bias correction

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t, \quad v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2$$

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t}, \quad \hat{v}_t = \frac{v_t}{1-\beta_2^t}, \quad \theta_{t+1} = \theta_t - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \varepsilon}$$

Watch for. Bias correction explicitly on. Report $\alpha, \beta_1, \beta_2, \varepsilon$.

Adam without bias correction Same as Adam baseline but uses m_t, v_t directly without hat terms. *Watch for.* Clear disclosure of bias correction off and a sanity check on a toy problem if they reimplemented.

Adam with $\beta_1 = 0$ (similar to RMSProp)

$$m_t = g_t, \quad v_t \text{ as above}, \quad \theta_{t+1} = \theta_t - \alpha \frac{g_t}{\sqrt{v_t} + \varepsilon}$$

Watch for. This behaves similar to RMSProp but is not identical. Report ε and any decay specifics.

Adam with decoupled weight decay (AdamW)

$$\theta_{t+1} = \theta_t - \alpha \left(\frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \varepsilon} \right) - \alpha \lambda \theta_t$$

Key distinction. Decoupled weight decay is an explicit parameter shrink step, not the gradient of an L2 penalty added to the loss. In Part three students must not switch to Adam with decoupled weight decay.

Optimizer grading checks

One validation loss threshold per dataset used across all methods. Sensitivity heatmaps for Adam family over learning rate and beta one, and learning rate and beta two, use equal budgets per grid point. Validation trajectories show median and interquartile range over the same seed set. Batch size and schedules are consistent unless justified. Warm up and accumulation steps are counted in gradient evaluation totals.

Regularization techniques

L2 weight decay (coupled) Add $\lambda \|\theta\|_2^2$ to the loss and differentiate through it. This is different from decoupled weight decay used in Adam with decoupled weight decay. *Watch for.* Whether biases and normalization parameters are excluded. The choice must be applied consistently across conditions.

Early stopping Stop when validation loss fails to improve by `min_delta` for a fixed patience, with the same maximum epoch budget across conditions. *Watch for.* Early stopping must not secretly change compute for some methods only. State if and when it triggered.

Dropout Randomly zero activations during training and scale at inference. *Watch for.* Exact placements and rates. Avoid dropout immediately before Batch Normalization. For convolutional layers, Spatial Dropout is often more appropriate. In evaluation, dropout must be off.

Target smoothing or noise For classification, label smoothing decreases over confidence and can improve calibration. For regression, small target noise can reduce variance. *Watch for.* Magnitude is stated. Applied on training loss only. If they make calibration claims, a reliability diagram or Expected Calibration Error is appropriate.

Modality appropriate augmentation or input noise *Images.* Flips, crops, jitter kept modest. *Tabular.* Input noise or Mixup for classification. *Text or sequence.* Small token dropout or mild noising. *Watch for.* No augmentation on validation or test. No label altering transforms without justification.

Regularization grading checks

Part three uses standard Adam with the exact winning hyperparameters from Part two. Baseline with no extra regularization, best single regularizer, and best combination are all trained under identical budgets and reported with dispersion across seeds. Placements and values are explicit in captions or text.

Accounting and comparability

What counts as one function evaluation One full pass computing the validation objective for the current candidate parameters, even if accumulated over micro batches.

What counts as one gradient evaluation One backward pass that produces gradients for a parameter update at the chosen batch size. If they use gradient accumulation, count the underlying micro batch backward passes.

Wall clock Reported on the same hardware class. If the hardware changes, results must be separated and clearly labeled.

Seeds, splits, and initialization The same random seeds, the same initialization, and the same data splits must be used across methods within a comparison. Any deviation must be justified and logged.

Quick grader checklist

- Randomized Optimization touches only the last one to three layers with at most approximately fifty thousand parameters, and uses equal function evaluation budgets across algorithms.
- Optimizer ablations use one validation loss threshold per dataset and report time or updates to reach it, with stability bands over the same seed set.
- Regularization study keeps the optimizer fixed to standard Adam with the winning Part two hyperparameters, reports baseline, best single, and best combination under identical budgets, and includes precise placements and values.
- All figures have readable axes at one hundred percent view and captions state the key settings that define the comparison.
- Compute accounting is present for every condition: number of gradient evaluations or function evaluations, and wall clock time.

Rubric

Total Points: 100 + 5 Extra Credit

Notes on Partial Credit

For this term, I'd like for all of us to use the 'all or nothing' mentality for grading. This worked quite well over the last couple semesters. This means if a section is worth 2 points, we will either award 2 points or 0 points (rather than 1 point for potentially getting close or just slightly missing it). This will help us stay more consistent throughout the grading cycle. Please reach out if you have any questions.

Structured Feedback – CS7641 Fall 2025 Report

Continuing from the Hypothesis Report Practice and SL Report, I want to put the point values by the section feedback. I have an example document I would like everyone to use as a template. You will still need to add individualized feedback for each student. Please reach out if you have any questions!

Datasets for Fall 2025

We require the students to use these two specific datasets. If they are not explicitly using these datasets, they will receive a zero. This should hopefully not be the case but please make a note on the sheet in the comments. For further information on these datasets, please see the SL Report description or the Kaggle repos.

- **Hotel Booking Demand**: Kaggle Repository: *Hotel Booking Demand*
- **US Accidents**: Kaggle Repository: *US Accidents*

1. Requirements (10 points)

- **7 points total:** This is the DOCSTRING-GTusername file on Canvas. All requirements met, full reproducibility ensured.
  - **2 points:** Overleaf and LaTeX used for report. Please double check the report exists.
  - **2 points:** Github Hash. Please double check that this hash works.
  - **2 points:** Instructions to run code. The can be anything but needs to be instructions to help with packages or init to run the code. If the instructions are on Github and not the DOCSTRING, please remove these 2 points.
  - **1 point:** The document is a PDF.
- **1 point:** Report is 8 pages or less. If there is anything more, please make a note to the student for future reports.
- **1 points:** Graphs and text are legible. This means you do not need to do any zooming while on Canvas at 100%.
  - This point can be hard. Many times the student will have some cropped graphs that are hard to read at 100% zoom on a computer. If there are 3 or more graphs that are hard to read the axis or key/legend, please deduct this point.
- **1 points:** Proper References. This is a check to make sure their reference are consistent in formatting. APA, MLA, or IEEE are fine, but need to stay consistent. IEEE does have a website url in the description. This point should be taken off if the student includes something outside the format chosen or text that should be included in the main body.
  - Additionally, if the student does not have a citation outside of the course material or citing the environment, you should take off a point. The rationale here is that the student should be looking for connections to the literature to support their results.

2. Hypothesis (10 pts)

- **4 points:** Initial Hypothesis
  - (**3 points**) For stating an explicit hypothesis for what the student expects in the report. This can be a wide range of potential hypotheses. The report does not need to have a hypothesis for each section, rather these sections should help develop an overall narrative to keep the paper focused.
  - (**2 points**) Explicitly stating evidence to support the claim of the hypothesis from a lecture, paper, or generally known theory. The hypothesis must be grounded in evidence before exploration. Since both datasets are quite messy, this may be from basic EDA while exploring the date.
- **4 points:** Follow Through with Hypothesis
  - (**2 points**) Following up with the hypothesis in their discussion or conclusion. It is not enough to just state a hypothesis but follow through from a narrative perspective needs to be met at the end of the report. This should be in their discussion section or further on in their results.
  - (**3 points**) Providing direct evidence from their report to support if the hypothesis has been met or not. This can be short, however there needs to be something explicit to help better summarize. The student should receive full credit here if they mention hard numbers for comparison (since this is more relative with the subscribed datasets).

3. Part 1: Randomized Optimization on the last several layers (24 points)

- **6 points: Algorithm coverage and disclosures (two points for each algorithm).** All three required algorithms are run on both datasets under the same layers, parameter cap, and evaluation budgets. Full, reproducible disclosures:
  - **Randomized Hill Climbing (2 points):** restart policy and step size schedule; perturbation or neighborhood operator (distribution and scale) and any adaptation rule.
  - **Simulated Annealing (2 points):** initial temperature, cooling schedule, and any step size cooling; perturbation operator (distribution and scale) and adaptation rule.
  - **Genetic Algorithm (2 points):** population size, selection method, crossover operator, mutation rate, and whether elitism is used; representation (real coded) and variation design.
- **4 points: Budget fairness and accounting.** Equal function evaluation budgets across Randomized Hill Climbing, Simulated Annealing, and Genetic Algorithm. If a plateau rule is used, it is identical across algorithms and attempted evaluations still count. For each run, report the number of function evaluations (every validation objective counts as one) and wall clock time on the same hardware class. In addition, the Randomized Optimization scope and objective hygiene are satisfied: only the last one to three layers are tuned; total trainable parameters for Randomized Optimization are at most approximately 50,000 with the exact count reported; the same layers and the same parameter cap are used across all Randomized Optimization algorithms; the objective is the full validation loss computed with `model.eval()` (dropout disabled and Batch Normalization uses stored running statistics); no gradient steps are interleaved with Randomized Optimization. Any over budget runs are clearly marked and excluded from direct claims.
- **3 points: Progress and diagnostics.** Best so far objective versus function evaluations curves (logarithmic x axis allowed) per algorithm and per dataset with readable captions listing operator settings. Failures (not a number or infinity) are counted as evaluations and failure rates are reported. Acceptance rates, where applicable, or analogous diagnostics are summarized.
- **3 points: Perturbation and variation design.** Layer aware proposal scales, for example by tensor root mean square or fan in, or by the initialization standard deviation, to avoid domination by large magnitude tensors. Bounds, clipping, or constraints, if any, are stated and kept constant across algorithms. A real coded representation for the Genetic Algorithm, or a justified alternative, is used and documented.
- **2 points: Initialization and seeds.** The initial checkpoint policy is stated, for example the Supervised Learning Report backbone or the model trained in Part 2. Warm start is applied consistently across Randomized Hill Climbing, Simulated Annealing, and Genetic Algorithm. Random seeds are fixed and logged; identical initial weights are used across algorithms for a given dataset and architecture.

- **2 points: Results table and winner callout.** For each dataset, a comparison table lists for Randomized Hill Climbing, Simulated Annealing, and Genetic Algorithm: best validation loss, test metric, number of function evaluations, and wall clock time. The winner is identified with a brief, mechanism level explanation, for example Simulated Annealing exploration versus Randomized Hill Climbing locality or Genetic Algorithm recombination benefits.
- **4 points: Comparative discussion of Randomized Optimization efficacy.** A focused discussion that answers: Did Randomized Optimization help in practice relative to the Supervised Learning Report backbone and to gradient based training in Part 2 under matched budgets? Include a compute aware comparison that weighs any improvement in the test metric against the additional function evaluations and wall clock time. Use paired per seed deltas to show consistency, identify the conditions under which Randomized Optimization helped or did not help, and provide a mechanism level explanation grounded in the reported diagnostics. State a clear takeaway per dataset about when Randomized Optimization is worth the cost.

4. Part 2: Adam ablations on the full network (26 points)

- **5 points: Coverage and fairness.** All seven optimizers are run on both datasets with the same splits, batch size, and hardware class: Stochastic Gradient Descent without momentum, Stochastic Gradient Descent with momentum, Nesterov momentum, Adam baseline, Adam without bias correction, Adam with β_1=0 (similar to RMSProp), and Adam with decoupled weight decay (AdamW). Adam baseline is explicitly included. Budgets are matched across methods. If any are missing, subtract a point.
- **3 points: Time or steps to threshold.** A single validation loss threshold ℓ is defined once per dataset and used across all optimizers. Plots and tables show time or update count to reach ℓ, including an explicit greater than budget indicator for failures.
- **3 points: Sensitivity heatmaps.** Coarse grids over (α, β_1) and (α, β_2) for Adam family methods, with identical budgets per grid cell and the same data split. Divergent or brittle regions are shown.
- **3 points: Stability.** Validation trajectories report median and interquartile range over three to five seeds. Divergence and not a number rates are noted. Seeds are fixed and logged.
- **3 points: Generalization gap.** Training vs. validation curves at the chosen budget and a concise discussion of overfitting or underfitting behavior for each optimizer.
- **2 points: Budget accounting.** Gradient evaluation counts including warm up and accumulation and wall clock time are reported for each run. Any over budget runs are clearly marked and excluded from direct claims.
- **7 points: Comparative discussion of results and decision rule.** A focused synthesis that:
  - **2 points:** Integrates evidence across optimizers and datasets, relating speed to threshold, final test performance, stability across seeds, and sensitivity map structure.
  - **2 points:** Provides a compute aware comparison, for example a frontier style narrative that weighs test metric against total compute, and explains trade offs.
  - **2 points:** Attributes observed differences to mechanisms with citations to figures, for example the effect of bias correction on early progress or the role of momentum in stability.
  - **1 point:** States a clear decision rule per dataset for which optimizer to favor under a fixed budget and identifies the standard Adam hyperparameters that will be used in Part 3, with justification.

5. Part 3: Regularization study (24 points)

- **1 point: Optimizer scope discipline.** Standard Adam only with the best Part 2 hyperparameters (learning rate, β_1, β_2, ε, batch size). No optimizer retuning or variant substitution in Part 3.
- **5 points: Required regularizers attempted.** All five required regularizers are attempted: L2 weight decay (coupled), early stopping, dropout (placements documented), target smoothing or noise, and modality appropriate augmentation or input noise.
- **5 points: Baseline vs. best single vs. best combination.** For each dataset, shows (i) baseline Adam with no additional regularization, (ii) best single regularizer, and (iii) best combination, all under identical training budgets. The winner is identified clearly.

- **4 points: Precise disclosures.** Exact regularizer values and placements (for example, dropout layers), L2 coupling and any exclusions (bias or normalization parameters if excluded), early stopping rule (patience and `min_delta`), and augmentation policy summary. No optimizer retuning in Part 3.
- **6 points: Analysis and comparison quality.**
  - **2 points: Evidence with dispersion** — Test set metrics are reported with seed dispersion (median and interquartile range; mean and 95 percent confidence interval if five or more seeds). If a method improves validation but hurts test, it is reported and explained. Paired per–seed deltas are shown when comparing the best single regularizer to the best combination.
  - **2 points: Head–to–head validity** — All comparisons are budget matched with the same seeds, splits, batch size, and hardware class. Training and evaluation modes are correct (dropout disabled, Batch Normalization uses running statistics in evaluation). No augmentation is applied on validation or test.
  - **2 points: Compute accounting** — Gradient evaluation counts (including warm up and accumulation) and wall clock time are logged for every condition. Any over budget runs are clearly marked and excluded from direct claims.
- **3 points: Regularization sweep and mechanisms.** A figure or table compares individual regularizers and the best combination. A short, mechanism level explanation (for example, smoothing, calibration, capacity control) is tied to observed changes.

6. Conclusion (6 points)

Note: The conclusion may be its own section, or you may include per–dataset conclusions with a final overall comparison across datasets and algorithms.

- **6 points (Full credit).** Clear, synthetic wrap-up that:
  - Explicitly discusses Type I (FP) and Type II (FN) errors for the chosen targets and ties them to operating points (classification) or residual structure (regression).
  - Presents at least three evidence-backed takeaways that compare algorithms and datasets (e.g., accuracy/PR-AUC vs. wall-clock, stability vs. capacity, sensitivity to hyperparameters/features).
  - Connects results to course concepts (generalization/Occam, cross-validation choice, bias–variance, appropriate error metrics such as MAE/MSE or PR-AUC).
  - Acknowledges limitations (e.g., class imbalance, potential leakage risks, data shifts) and names one concrete next step.
  - Is consistent with reported figures/tables and cites them directly.
- **3 points (Partial credit).** A conclusion is present but misses three or more elements above: Type I/II noted only superficially; fewer than three concrete takeaways; weak or non-specific links to course concepts; limited comparison across datasets/algorithms; minimal justification from figures/tables.
- **1 point (Bare mention).** No dedicated conclusion; only a brief remark embedded elsewhere (e.g., "SVM performed best") with little or no evidence, no discussion of Type I/II, and no connection to course ideas.
- **0 points (No conclusion).** Absent, off-topic, or statements that contradict reported results without justification.

Extra Credit: Part 4: Integrated Best Combination (Racked: 5, 3, 1, or 0 points)

- **5 points:** The final results includes all three required components: (1) standard Adam with the best Part 2 hyperparameters (same batch size), (2) the best regularization combination from Part 3 with exact values and placements, and (3) Randomized Optimization fine tuning on the same last one to three layers and the same parameter cap as Part 1 using the best performing Randomized Optimization algorithm. Randomized Optimization uses at most ten percent of the Part 1 function evaluation budget and at most five small interaction trials; seeds, hardware class, and data splits are unchanged. The submission includes a final comparison table with mean and ninety five percent confidence interval over three to five seeds for all four pipelines (Adam best with no regularization; Adam plus best single regularizer; Adam plus best combination; Adam plus best combination plus Randomized Optimization fine tuning), a compute frontier

plot showing test metric vs. total compute, paired per seed improvement deltas, full compute accounting (gradient evaluations, function evaluations, wall clock time), and a concise hypothesis resolution with a clear mechanism level explanation.

- **3 points:** All three components are present as above, and Randomized Optimization budget limits are respected, but one or two presentation or analysis elements are missing or weak (for example, confidence intervals omitted, compute frontier absent, no paired per seed deltas, or incomplete compute accounting). Seeds, hardware class, data splits, and budgets remain consistent. A brief hypothesis resolution is present with numerical evidence.
- **1 point:** An attempt at integration is made, but a major requirement is missing or violated: for example, Randomized Optimization fine tuning is omitted or uses different layers or a different parameter cap than Part 1, Adam hyperparameters are not the best from Part 2, the Randomized Optimization budget exceeds ten percent without disclosure, or comparisons are not budget matched. Evidence is limited to single seed or best seed only, or there is no clear compute accounting. Any hypothesis discussion is superficial.
- **0 points:** Part 4 is not attempted.

Total: 100 points + 5 Extra Credit

Version Control

- v1.0 - 08/20/2025 - TJL created and wrote the OL Report Rubric.