OPTIMIZATION / LEARNING THEORY / REPRODUCIBLE BENCHMARKS

Free-Energy Allocation for Sample-Efficient Neural Training

Variance, entropy, and speedups from random initialization

Anonymous · November 2026

Prepared for Agentic Meta-Journal

Abstract

We develop Free-Energy Allocation Sampling, a minibatch optimizer that minimizes a variance-based energy while rewarding Shannon entropy of the allocation. The exact integer allocation uses a finite greedy solver. The resulting stratified gradient is conditionally unbiased for the original empirical loss and requires no additional scoring pass. We prove gradient-exposure speedups from random initialization for nonlinear ReLU training with replicated strata and for a nonduplicate finite regression problem with a uniform-sampling lower bound. A rare-label construction yields a separate information-theoretic bound for adaptive algorithms that query records uniformly. The implementation is compared with full gradient descent, uniform minibatch SGD and Adam, and a fixed-stratification control on 26 introductory machine-learning and AI tasks. Sample counts, index-adjusted counts, optimization time, and held-out performance are recorded separately. FEAS-Adam reached the validation target on 24 of 26 tasks and full GD on 11. Against uniform Adam, it used fewer gradient examples on 7 of 23 tasks reached by both. In the checkpointed suite FEAS-Adam used fewer examples and seconds than uniform Adam on the three CNN benchmarks. Exact-crossing replays distinguish exposure savings from the wall-clock effects of vectorization, step size, and per-update overhead. A separate experiment measures the gains and costs of batch and step calibration.

1 The training problem

The spacetime manuscript [1] writes nonequilibrium free energy as energy minus temperature times entropy, with an exact relative-entropy decomposition. It also makes the accounting boundary explicit. We transfer both ideas to minibatch training. A batch allocates a finite number of gradient queries across data strata. Its energy is a variance cost, and its entropy measures how broadly that budget is distributed. The accounting boundary counts the examples consumed by a run and separately records the cost of preparing its stratum index.

Let $S = \{1, \dots, N\}$ be a finite training set, partitioned as $S_1, \dots, S_m$. The empirical objective and gradients are

$$L(\theta) = \frac{1}{N} \sum_{i=1}^{N} \ell_i(\theta), \qquad g_i(\theta) = \nabla \ell_i(\theta), \qquad g(\theta) = \nabla L(\theta).$$

We use differentiable points or a fixed convention for the derivative at a kink. Write $\pi_j = |S_j|/N$, $\mu_j = \mathbb{E}[g_i \mid i \in S_j]$ and $v_j = \mathbb{E}[\|g_i - \mu_j\|^2 \mid i \in S_j]$, with expectations over uniform records within the stratum. The entropy term acts on the allocation of training queries. The model objective L is the same objective seen by the gradient descent baselines. Energies in the allocation have the units of squared gradients or of the chosen proxies, and the analysis concerns computational exposure counts and measured seconds.

Neyman's optimal allocation [2] and subsequent work on importance sampling for stochastic optimization [3, 4, 5, 6] motivate reducing gradient variance before taking a step. With a fixed compute budget, stratified sampling additionally removes the variance caused by random variation in which strata are represented.

Our specific construction combines positive integer counts, a relative-entropy penalty, a predictable proxy that reuses the current forward pass, and an auditable comparison with both fixed stratification and uniform sampling. Variance-reduced methods that maintain a gradient reference, such as SVRG [7], use a different information budget. We make no novelty claim for Neyman allocation, importance correction, or Adam itself [8].

2 Free-energy allocation

Fix a batch size $B \geq m$. Choose $k_j \geq 1$ with $\sum_j k_j = B$ and put $q_j = k_j/B$. Conditional on these counts, draw k_j independent uniform records from S_j with replacement. The weighted gradient and allocation energy are

$$\widehat{g} = \sum_{j=1}^{m} \frac{\pi_j}{k_j} \sum_{s=1}^{k_j} g_{I_{j,s}}, \tag{1}$$

$$\mathcal{V}_v(q) = \sum_{j=1}^{m} \frac{\pi_j^2 v_j}{q_j}. \tag{2}$$

For the uniform reference $u_j = 1/m$, solve

$$\min_{\substack{k_j \in \mathbb{Z},\ k_j \geq 1 \\ \sum_j k_j = B}} \Phi(k/B) = \mathcal{V}_{\widehat{v}}(k/B) + \tau D_{\mathrm{KL}}(k/B \| u), \qquad \tau \geq 0. \tag{3}$$

Here $\widehat{v}$ is a nonnegative predictable variance estimate. Since $D_{\mathrm{KL}}(q\|u) = \log m - H(q)$, this is a direct energy–entropy tradeoff. Positive counts give every stratum a draw in each update. They also bound the amplification of a record's gradient by π_j, even when the fitted score in its stratum is zero. The corrected batch objective is the same weighted sum of per-example losses as in the gradient formula. For positive temperature, an optimum of the combined objective maximizes entropy among allocations at or below its variance energy, and minimizes energy among allocations at or above its entropy. Any strict improvement in either constrained problem would lower the same combined objective.

Theorem 1 (Conditional variance ledger)**.** *If the counts are measurable with respect to the history before the draws, then*

$$\mathbb{E}[\widehat{g} \mid \theta, k] = g, \qquad \mathbb{E}[\|\widehat{g} - g\|^2 \mid \theta, k] = B^{-1}\mathcal{V}_v(q).$$

An iid uniform minibatch of size B has variance

$$\frac{1}{B}\left(\sum_j \pi_j v_j + \sum_j \pi_j \|\mu_j - g\|^2\right).$$

Proof. Each within-stratum mean estimates μ_j, so the corrected expectation is $\sum_j \pi_j \mu_j = g$. The centered within-stratum means are independent conditional on the past. Their covariance traces are v_j/k_j, which gives the second identity by summation with weights π_j^2. For an iid draw, write $g_i - g = (g_i - \mu_j) + (\mu_j - g)$. Conditional on j, the cross term has expectation zero. Divide the resulting trace by B for the iid average. □

Thus proportional allocation $q = \pi$, when feasible exactly, removes the between-stratum term. Minimizing $\mathcal{V}_v$ alone over the simplex without count constraints gives $q_j \propto \pi_j \sqrt{v_j}$ for positive variances by Cauchy–Schwarz. The entropy term moves this allocation toward a broader distribution. If all v_j vanish, the integer solution maximizes entropy subject to the count constraints and the gradient is exact for every feasible allocation.

Theorem 2 (Exact integer allocation)**.** *Starting with one draw in each stratum, assign the remaining $B - m$ draws by selecting the largest successive decreases of*

$$\phi_j(k) = B\pi_j^2 \widehat{v}_j/k + \tau(k/B)\log(mk/B).$$

The resulting counts globally minimize (3)*.*

Proof. Each ϕ_j is convex for $k > 0$. Hence $\phi_j(k) - \phi_j(k+1)$ is nonincreasing in k. Any feasible solution selects an initial segment of the decreases in each stratum. Sort the first $B - m$ possible decreases of every stratum together, choosing earlier decreases first within a stratum in the event of a tie. The largest $B - m$ decreases form feasible initial segments. No other set of $B - m$ decreases has a larger sum. The starting objective minus this sum is the desired minimum. □

The continuous relaxation has an exact ledger of its own. Let $a_j = \pi_j^2 \widehat{v}_j$, $q^\star$ minimize Φ on $q_j \geq 1/B$, and let q be feasible. Then

$$\Phi(q) - \Phi(q^\star) = \langle \nabla\Phi(q^\star), q - q^\star \rangle + \sum_j \frac{a_j(q_j - q_j^\star)^2}{q_j(q_j^\star)^2} + \tau D_{\mathrm{KL}}(q\|q^\star). \tag{4}$$

Subtract the linearization of a_j/q_j to obtain the rational remainder. Subtract the linearization of $\sum q_j \log(q_j/u_j)$ to obtain $D_{\mathrm{KL}}(q\|q^\star)$, using $\sum_j (q_j - q_j^\star) = 0$. The first term is nonnegative by convex optimality, and is zero for a simplex-interior optimum away from the floors. This gives the same type of relative-entropy accounting used in the source manuscript, with an additional nonnegative remainder for a nonlinear energy. In particular $\tau > 0$ gives uniqueness in the continuous problem. The discrete optimizer is furnished by theorem 2.

For an interior solution and $\tau > 0$, the stationarity equation can also be written as an implicit Gibbs law

$$q_j^\star \;\propto\; u_j \exp\!\left(\frac{a_j}{\tau(q_j^\star)^2}\right).$$

Equivalently, $\sum_j a_j/q_j = \sup_z \sum_j (2\sqrt{a_j} z_j - q_j z_j^2)$, from completing the square. For fixed z, the resulting allocation energy is linear in q, with state energy $-z_j^2$. The stationary law is conditional on a current marginal variance cost. The discrete solver avoids iterating this fixed point.

2.1 Implementation and descent

The executable starts with positive counts nearest $B\pi$ in squared error. From completed batches it estimates the trace of the within-stratum covariance of a cheap output-space residual. For cross-entropy this is the softmax probability vector minus the one-hot target. For multilabel gates it is a sigmoid residual. For squared loss it is the prediction residual. For the margin loss it is the active signed margin derivative, and for the two clustering objectives the per-record loss is the scalar proxy. The head proxy measures output-level variation. The true ledger uses v_j in parameter space, which can differ when features, Jacobians, or label distributions vary. Statistics use sample variances when $k_j > 1$, an exponential moving average with decay 0.85, and the last value when only one sample is available. Every four completed draws the solver refreshes the counts. It uses $\tau = 0.06 \max(10^{-12}, \sum_j \pi_j \widehat{v}_j)$. The initial counts and all subsequent counts are chosen before the records in the corresponding batch are drawn.

The step $\theta^+ = \theta - \eta D\widehat{g}$, for predictable positive definite D, minimizes the estimated linearized loss plus the quadratic step cost $\|\theta^+ - \theta\|_{D^{-1}}^2/(2\eta)$. For $D = I$ the descent lemma for an L_s-smooth loss yields

$$\mathbb{E}[L(\theta^+) \mid \theta, k] \leq L(\theta) - \eta(1 - L_s\eta/2)\|g\|^2 + \frac{L_s\eta^2}{2B}\mathcal{V}_v(q).$$

Insert $\widehat{g}$ into the upper quadratic approximation and use theorem 1. Under a local Polyak–Łojasiewicz inequality $\|g\|^2 \geq 2\mu(L - L^\star)$ and $\eta \leq 1/L_s$, the conditional expected gap contracts by at least the factor $1 - \mu\eta$ plus the displayed variance term wherever the hypotheses hold. This is a variance-sensitive descent estimate for FEAS with SGD. In the Adam variant the same unbiased batch gradient is passed to Adam's moment recursions. The nonlinear moment step is analyzed empirically.

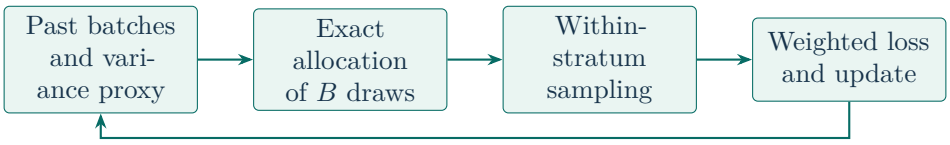


Figure 1: The current residual updates the allocation history after the weighted gradient is formed. The feedback affects later batches.

Proposition 3 (A sufficient condition for a variance gain)**.** *Let $q^\star$ minimize the fitted discrete objective, let q^0 be any feasible reference, and let $e_B = B\sum_j \pi_j^2|\widehat{v}_j - v_j|$. Then*

$$\mathcal{V}_v(q^\star) \leq \mathcal{V}_v(q^0) + \tau[D_{\mathrm{KL}}(q^0\|u) - D_{\mathrm{KL}}(q^\star\|u)] + 2e_B.$$

If $q^0 = \pi$ is feasible, the true variance is strictly smaller than for uniform iid sampling whenever the between-stratum term in theorem 1 exceeds $\tau D_{\mathrm{KL}}(\pi\|u) + 2e_B$.

Proof. Every feasible q has $q_j \geq 1/B$, so replacing v by $\widehat{v}$ changes $\mathcal{V}$ by at most e_B. Compare the fitted objective at its minimizer and at q^0, and apply the error bound at both. With $q^0 = \pi$, $\mathcal{V}_v(q^0) = \sum_j \pi_j v_j$. Divide by B and use theorem 1 and nonnegativity of relative entropy. □

A count floor, a stable exact allocation solver, and no access to future batch data suffice for unbiasedness. Variance improvement additionally depends on the partition and proxy. If all stratum means agree and the within-stratum variances and masses are homogeneous, fixed proportional counts and uniform iid have the same variance. In that case a more expensive sampler can take longer for the same statistical progress.

3 Speedups from random initialization

We distinguish a gradient-example oracle from a complete data acquisition procedure. One gradient-example evaluation reads one record and forms its contribution at the current iterate. A stratum index can be available as metadata, or formed with one extra pass. The following constructions give explicit comparisons with scan-based full gradient descent and with uniform minibatch sampling.

Theorem 4 (Exact trajectory compression for a nonlinear ReLU network)**.** *For any integers $m \geq 1$ and $N \geq m$, there is a finite N-record ReLU training problem with m strata in which a* FEAS *batch of size m yields the exact full gradient at every update. There is a centered random initialization under which the loss converges geometrically almost surely. Standard independent Gaussian initialization has positive probability of the same guarantee. For t updates the two gradient-example counts are mt and Nt, respectively. If a full pass constructs the index, the count for* FEAS *is at most $N + mt$.*

Proof. Each stratum j repeats the same input e_j and target 1. Use $2m$ ReLU units with tied input and output weights, whose prediction on e_j is

$$f_\theta(e_j) = \theta_j(\mathrm{ReLU}(\theta_j) - \mathrm{ReLU}(-\theta_j)) = \theta_j^2.$$

For example the paired hidden weights are $\theta_j e_j$ and $-\theta_j e_j$, and the corresponding output weights are $\theta_j, -\theta_j$. All predictions at the other input coordinates vanish in that pair. The square loss is $L(\theta) = \sum_j \pi_j(\theta_j^2 - 1)^2/2$.

Within every stratum the loss and gradient are identical across records, so a batch of one draw per stratum with the correction π_j is exact. Pair the initialization and step policy of the scan-based method with those of FEAS. The iterates agree inductively.

Take independent random magnitudes $r_j \sim \mathrm{Unif}[1,2]$ and independent fair signs, so $\theta_{0,j} = \pm r_j$. The objective is even in each coordinate. On the positive band $r \in [1,2]$, the weighted component loss has second derivative $\pi_j(6r^2-2) \in [4\pi_j, 22\pi_j]$. With $0 < \eta \le 1/(22\max_j \pi_j)$, the mean value theorem gives

$$0 \le r_j^+ - 1 \le (1 - 4\eta\pi_j)(r_j - 1).$$

Indeed the derivative at 1 is zero, so the update cannot cross 1 and cannot increase the magnitude. The negative band is invariant by symmetry. This gives geometric convergence of the loss. For independent standard Gaussian coordinates the event $1 \le |\theta_{0,j}| \le 2$ for all j has probability $[2(\Phi_{\mathrm{N}}(2) - \Phi_{\mathrm{N}}(1))]^m > 0$. The evaluation counts follow from the two sampling rules. Including the index pass preserves an advantage whenever $t(N-m) > N$. □

The exact compression property holds for any architecture with losses that are identical within each known stratum. The example above establishes a nonconvex ReLU parameterization and a random basin with a closed convergence bound. The same phenomenon occurs with independent trainable input and output weights.

Proposition 5 (Untied two-layer random basin)**.** *Use the same records with predictions $f(e_j) = v_j\,\mathrm{ReLU}(u_j)$. If the u_j, v_j are independent uniform random variables on $[1,b]$, $b > 1$, and $\eta\max_j \pi_j \le 1/(4b^2)$, both methods in theorem 4 attain the same geometric convergence of squared loss almost surely. The box also has positive probability under independent Gaussian initialization of all these parameters.*

Proof. On a fixed stratum put $p = uv$ and $h = \eta\pi_j$. In the active region, $u^+ = u - h(p-1)v$, $v^+ = v - h(p-1)u$. If $u, v \le b$ and $p \ge 1$, then $u, v \ge 1/b$, and the relative decrease of u is $h(p-1)v^2/p \le hb^2$. The same bound holds for v, so both stay positive and at most b. Direct multiplication gives

$$p^+ - 1 = (p-1)(1 - h(u^2+v^2) + h^2 p(p-1)).$$

The bracket is at least $1 - 2hb^2 \ge 1/2$. It is at most $1-h$, since $u^2+v^2 \ge 2p$, $h(p-1) \le 1/4$, and $p \ge 1$. Hence $p^+ \ge 1$ and the output error contracts. This invariant proves the stated convergence. The one-draw-per-stratum gradient remains exact for both independent parameters. □

The companion executes the tied example with $N = 8{,}192$, $m = 4$, five random seeds, and loss target 10^{-6}. It checks equality of the two trajectories numerically to double precision. The raw gradient-example ratio was 2,048 to one in every run, and the median ratio including one full index pass was 28.60 to one. The stratified implementation was faster in 4 of 5 runs, with a median ratio of 1.23 on these short CPU timings. The full-gradient comparator actually evaluates each record at each step. A solver that first compresses the duplicated records uses the same sufficient information as the stratified gradient. The exact separation explains the benefit of avoiding repeated gradient evaluation in the scan-based comparator.

Theorem 6 (A nonduplicate exposure separation against uniform SGD)**.** *Let a trainable scalar output bias fit a finite regression population by minimizing $(\theta - Y)^2/2$. Let there be m balanced known strata with the same positive within-stratum variance σ_w^2, and let their conditional means have variance σ_b^2. Write $\theta^\star = \mathbb{E}Y$, $\sigma^2 = \sigma_w^2 + \sigma_b^2$, and take a random initialization independent of the draws with $0 < M = \mathbb{E}(\theta_0 - \theta^\star)^2 < \infty$. A single unit step using B FEAS draws, with $m \mid B$, achieves mean squared parameter error σ_w^2/B. Any uniform-with-replacement minibatch SGD with a deterministic step schedule and n total draws has mean squared error at least*

$$\frac{M\sigma^2}{nM + \sigma^2}.$$

Consequently the asymptotic exposure ratio at a target $\varepsilon \downarrow 0$ is at least σ^2/σ_w^2, which can be arbitrarily large with no duplicate labels within a stratum.

Proof. At balanced masses and equal variances, the uniform allocation minimizes $\mathcal{V}$ by convexity and symmetry and also minimizes the KL term. It is feasible for $m \mid B$. The executable starts with the same counts. The unit step on the scalar quadratic outputs the stratified sample mean. Its variance is σ_w^2/B by theorem 1. For uniform SGD, expand the linear recursions. For deterministic step sizes and batch sizes the final estimator has the form $a\theta_0 + \sum_{\nu=1}^{n} w_\nu Y_\nu$, with $\sum_\nu w_\nu = 1 - a$ and iid Y_ν. Independence and zero means of the centered draws give squared error $Ma^2 + \sigma^2 \sum_\nu w_\nu^2$. Cauchy–Schwarz bounds this from below by $Ma^2 + \sigma^2(1-a)^2/n$, whose minimum over real a is the displayed expression. Hence any such estimator with error at most $\varepsilon < M$ requires $n \geq \sigma^2(1/\varepsilon - 1/M)$. The stratified count is at most $\sigma_w^2/\varepsilon + m$. Use disjoint strata of centered, nonconstant finite grids translated by distinct offsets to make σ_b^2/σ_w^2 as large as desired while keeping all records distinct. □

In the numerical example, the four offsets are $-6, -2, 2, 6$ and each stratum has a 101-point centered grid in $[-0.2, 0.2]$. The total-to-within variance ratio is 1,471.59. Across 4,096 simulated draws of 128 records, the observed squared errors were 0.15333 for the uniform mean and 0.000108 for the stratified mean. Independent normal initialization satisfies the theorem, and the unit-step estimate discards its initial bias. The simulation compares the stratified and iid sample means at equal exposures, and the paper's lower bound allows any deterministic step schedule of the stated SGD form. It does not apply to a sampler that explicitly requests a rare group or to an arbitrary nonlinear estimator.

Theorem 7 (An adaptive rare-label lower bound)**.** *There is a two-coordinate ReLU-feature model with $N - 1$ records in one stratum and one in another such that one* Feas *update with two draws learns both labels exactly from any initialization. Any algorithm that accesses the rare label only through iid uniform record queries has probability of a wrong rare-label sign at least $\frac{1}{2}(1 - 1/N)^n$ after at most n queries under a fair prior on that label.*

Proof. Use a linear trainable head on the fixed features $\mathrm{ReLU}(e_j) = e_j$. The common label is fixed and the rare label is independently ± 1. The conditional loss in stratum j is $(\theta_j - y_j)^2/2$. The balanced two-draw estimator gives the exact gradient since each stratum is internally identical. The diagonal preconditioner $D = \mathrm{diag}(\pi_j^{-1})$, which uses only known group masses, with step size 1 sets $\theta_j = y_j$. If no uniform query sees the rare record, all transcripts have identical distributions under the two rare labels, including any adaptive updates based on the observed data. The random initialization is independent of the unknown sign. At least half of the conditional decisions fail. The event of missing that record in n draws has the stated probability, which also lower bounds missing it under earlier stopping. □

To reduce the rare-label error probability below $\delta < 1/2$, the uniform sampler therefore requires at least $\log(1/(2\delta))/[-\log(1 - 1/N)]$ draws. The theorem charges the acquisition of the unknown label, assuming the groups are recognizable from the inputs. It gives a precise condition under which targeting an informative subset improves observation cost even for adaptive learners. In the main classification experiments the strata are usually formed from the known label pool, so this rare-label acquisition result and the measured gradient-example counts answer different questions.

3.1 When exposure savings imply time savings

If scan-based full GD uses t updates at total cost tNc_{full}, and Feas uses the same trajectory with index time I, m gradient examples per step of cost c_{mini} and step overhead a, the stratified execution is faster when

$$I + t(mc_{\mathrm{mini}} + a) < tNc_{\mathrm{full}}.$$

The per-example costs can differ because large batches make better use of vectorized kernels. At fixed data size this condition can fail despite fewer gradient examples. In the empirical suite we record elapsed optimization time at the same validation crossing as the example count, so the two claims can be checked independently.

4 Benchmark protocol

The benchmark suite covers the trainable examples in introductory ML and AI curricula. The taxonomy follows the regression, classification, sequence, clustering, recommendation, and reinforcement-learning topics in MIT 6.036 and Stanford CS229, and the search and decision topics in Stanford CS221 and Berkeley CS188 [9, 10, 11, 12]. For search and games, we train neural predictors from exact planning or minimax labels. For two control tasks, we fit an offline policy or value function. The suite gives task-by-task results within that scope.

Family	Examples and objectives
Regression and margins	Synthetic linear regression, a cubic feature map, diabetes regression, a hinge-loss linear margin model, and logistic breast-cancer classification.
Logic and classification	AND, OR, and XOR in one network, Iris and Wine softmax, nonlinear moons and circles, and an MLP on 8×8 digits.
Unsupervised objectives	A rank-eight linear autoencoder in the PCA setting, a nonlinear autoencoder, soft k-means, and a fixed-variance Gaussian mixture with learned means and weights.
Vision	MLP and CNN variants for MNIST, a Fashion-MNIST CNN, and a CIFAR-10 CNN.
Language and sequences	AG News topic classification on hashed word counts, and a GRU that predicts the majority bit of a length-12 sequence.
Recommendation	MovieLens 100K rank-12 matrix factorization with user and item biases.
Planning, games, and control	Eight-puzzle distance regression, tic-tac-toe minimax value classification, FrozenLake fitted action values, and CartPole imitation of a fixed linear policy.

For Iris, Wine, breast cancer, diabetes, and 8×8 digits we use the scikit-learn datasets [13]. The rank-eight model is a linear reconstruction objective, whose optimum selects a principal subspace under the classical linear autoencoder conditions [14]. Inputs for tabular linear and two-dimensional discriminative tasks are standardized with training statistics. Diabetes targets are standardized by the training target mean and standard deviation. The reconstruction tasks use pixels scaled to $[0, 1]$. For the two clustering problems, labels used to measure adjusted Rand index are omitted from the training loss and the stratum statistic. The soft k-means energy is $-s \log \sum_j \exp(-\|x - c_j\|^2/s)$ with $s = 0.35$. The mixture likelihood uses three isotropic components with standard deviation 0.55, plus learned centers and mixture weights.

MNIST and Fashion-MNIST supply 28×28 images, and CIFAR-10 supplies 32×32 color images [15, 16, 17]. We use capped, deterministic subsets, keep validation disjoint from training, and draw the test evaluation subset only from the official test split. The CNN has two convolution–ReLU–pooling blocks of widths 8 and 16. We use no batch normalization or data augmentation. AG News uses the published four-class splits [18], a 1,024-dimensional hashing map, and a single hidden layer of width 48. MovieLens 100K uses the official `u1.base`/`u1.test` separation [19]; the training and validation subsets are drawn from `u1.base`. The artifact provides source URLs and hashes of every downloaded file. The capped datasets and small models make a six-way CPU comparison feasible. Their accuracies should be read in the stated protocol.

The Boolean example trains on all four logical inputs and evaluates the exact joint truth table. The GRU sees a held-out split of distinct binary sequences. The eight-puzzle benchmark computes exact distances by breadth-first search from the solved board over all 181,440 reachable states, then partitions 12,000 distinct positions. Evaluation measures prediction error in moves. For tic-tac-toe, we enumerate legal nonterminal positions, label each with its minimax value, and split by position. These evaluations quantify prediction, with no claim about improvement in A* node expansions or online play.

The FrozenLake example fits the action-value function computed by value iteration with discount 0.99 on the standard slippery 4×4 map [20]. Its greedy policy is evaluated by exact 100-step goal-reaching probability. This evaluation uses the same 16-state model as the fit. CartPole uses state samples labeled by a fixed balancing policy and evaluates imitation on held-out states. A second CartPole metric averages

return across 24 paired reset states in a deterministic Euler simulator using the usual control task's constants and termination bounds [21]. Neither training process queries an online exploration environment. The comparison concerns optimization of fixed offline targets.

4.1 Pairing, tuning, and accounting

Each task compares six methods. GD forms the exact empirical gradient, accumulating chunks of 256 records where needed, and takes one update per full scan. SGD and Adam use iid uniform minibatches. Fixed-Adam uses the positive data-proportional allocation and the same corrected weights as Feas. Feas-SGD and Feas-Adam use the adaptive allocation in (3) and pass the weighted gradient to the named base optimizer. The uniform and stratified minibatch methods share the same batch size for a task. The supervised classifiers and margin model use label strata. The Boolean task uses input patterns, the rating task uses user-frequency quartiles, the puzzle uses Manhattan-distance bins, and FrozenLake uses a fixed state partition. Other nonclassification tasks use quantiles of the squared input norm.

A pilot with seed 913 picks the lowest terminal validation loss in each base learning-rate grid. The GD rates are $\{0.03, 0.12, 0.45, 1.0\}$, SGD rates are $\{0.025, 0.10, 0.30\}$, and Adam rates are $\{0.001, 0.004, 0.012\}$. The pilot budget is $\max(3N, \min(\lfloor C/3 \rfloor, 6144), 8B)$, where C is the main example budget. Only full steps fit inside the budget. The variants share the selected base optimizer rate. This makes the adaptive versus uniform and fixed comparisons informative without giving the adaptive method a separate tuning allowance. The pilot runs and their costs are in the archive.

The main trials use seeds 13, 37, and 71, with identical initial parameters across methods for each task and seed. Method order is independently permuted within each seed. Every task uses a fixed loss, threshold, batch size, data split, and gradient-example cap. Validation is measured at initialization and at checkpoints spaced by approximately $C/18$ gradient examples, rounded to the next eligible update. GD is evaluated only after full updates. The first observed crossing is an upper bound on the first crossing over all unobserved steps. We record the endpoint test score after the budget rather than choosing by test performance.

We record gradient-example exposures, the number of distinct records used for gradients, stratum-index visits, optimization seconds, and elapsed seconds including validation. The conservative index-adjusted count adds N visits for each stratified method. Timing includes model and optimizer setup, sampler construction, draws, forward and backward passes, variance-proxy calculations, and the update. Dataset materialization and validation are outside the optimization timer. CPU runs use one PyTorch thread per trial on an Intel Xeon Platinum 8573C. Methods for a task run sequentially, and different tasks can run simultaneously on distinct pinned cores. The recorded stack is Python 3.12.14, PyTorch 2.7.1 CPU, NumPy 2.3.5, and scikit-learn 1.8.0. Timing ratios describe this implementation and machine.

5 Measured results

The main suite contains 468 trials across 26 tasks. FEAS-Adam reached the validation target on 24 of 26 tasks and full GD on 11. Against uniform Adam, it used fewer gradient examples on 7 of 23 tasks reached by both. In the checkpointed suite FEAS-Adam used fewer examples and seconds than uniform Adam on the three CNN benchmarks. A target time is treated as right-censored when a run misses its target. The tables show medians over three seeds. The minimum possible median replaces each censored time by its terminal time. The upper bound replaces censored times by infinity. We show an interval when two seeds reached and these bounds differ, and mark the median as unreached if fewer than two seeds crossed. A zero exposure entry means that the initial model already met the validation criterion. Exposures count repeated draws each time, so a reduction in this column is a reduction in optimization queries. Organizing a labeled training pool by class uses labels that have already been acquired. Section 5.5 compares a smaller fixed label pool separately.

5.1 Paired method comparisons

The following aggregate uses tasks whose two median costs are observed for wins, ties, and losses. For exposure costs, the adjacent column counts tasks reached by the first method alone, the second method alone, or neither. The timing comparison uses positive median training exposures and point-identified times. The AG News Feas-Adam median is interval-identified and appears in the timing appendix. Positive finite ratios are the baseline median divided by the proposed median and then summarized across tasks.

Method pair	Cost	W/T/L	A/B/neither	Median ratio
FEAS-Adam / GD	Examples	5/2/3	14/1/1	1.74
FEAS-Adam / GD	With index	4/0/6	14/1/1	1.03
FEAS-Adam / GD	Seconds	1/0/7	—	0.14
FEAS-Adam / Adam	Examples	7/12/4	1/0/2	1.00
FEAS-Adam / Adam	With index	3/0/20	1/0/2	0.74
FEAS-Adam / Adam	Seconds	7/0/14	—	0.75
FEAS-Adam / Fixed-Adam	Examples	4/18/2	0/0/2	1.00
FEAS-Adam / Fixed-Adam	With index	4/18/2	0/0/2	1.00
FEAS-Adam / Fixed-Adam	Seconds	2/0/19	—	0.75
FEAS-SGD / SGD	Examples	3/14/2	1/0/6	1.00
FEAS-SGD / SGD	With index	3/0/16	1/0/6	0.72
FEAS-SGD / SGD	Seconds	1/0/16	—	0.64
FEAS-SGD / GD	Examples	9/2/0	9/0/6	1.67
FEAS-SGD / GD	With index	5/0/6	9/0/6	1.07
FEAS-SGD / GD	Seconds	1/0/8	—	0.19

On jointly reached tasks, FEAS-Adam saved gradient examples on 7 of 23 relative to uniform Adam, with 12 ties. After charging the index pass the win count was 3. Fixed stratification tied the adaptive allocation on 18 of 24 jointly reached tasks. The matching SGD comparison is similar, with most tasks tied in exposure count. In the checkpointed CNN runs FEAS-Adam used fewer examples and seconds than uniform Adam. MNIST CNN used 6,720 versus 11,136 examples and 1.17 versus 1.80 seconds. Fashion-MNIST CNN used 5,376 versus 8,064 examples and 1.33 versus 1.90 seconds. CIFAR-10 CNN used 7,008 versus 10,560 examples and 1.98 versus 2.86 seconds. The matched timing evidence is mixed across all tasks. Of 8 joint positive-exposure timings against full GD, FEAS-Adam was faster on 1. The eight-puzzle task is a joint example of exposure and timing gains against GD. The synthetic linear model is an example where the small full batch reached the target faster.

5.2 Timing audit with validation at every update

The checkpointed main suite can credit a method only at the next scheduled check, even if it crossed the target earlier. We audited all eight tasks with positive, observed median times for both GD and Feas-Adam. At each archived learning rate, we locate the first validation crossing by checking after every update. We replay the exact update sequence from the same seed to that endpoint with validation only before and after training. Every replay must reproduce its loss, metric, and update count. This separates measurement of the endpoint from execution of the timed path. We also audit the Adam-based methods on all three CNNs. After one warmup replay, we retain five replays per seed for the small tasks and three for the CNNs. Replays are run on a pinned CPU with one PyTorch thread on an AMD EPYC 9V74, distinct from the machine for the main suite. Each displayed time is a median over seeds of within-seed medians. Censored durations are used as lower bounds.

Vectorization and update counts. In the table below, entries are updates and optimization milliseconds to the same fixed target. GD (256) uses the original gradient accumulation. GD (one batch) uses

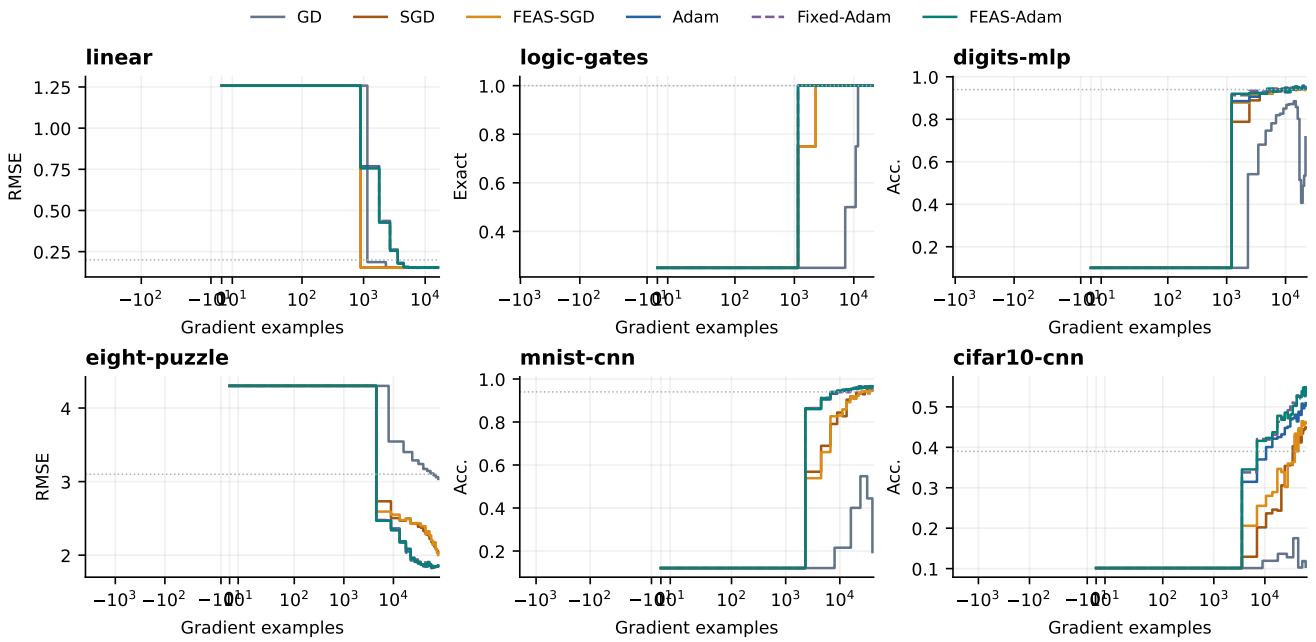


Figure 2: Median validation score versus gradient-example exposures for six illustrative tasks. Dotted lines mark the fixed target. The appendix reports all tasks. Axis scales separate the early updates from the larger budgets.

a single tensor forward and backward pass for the same empirical gradient, learning rate, and example budget. Both full-batch paths reached at the same update or were censored together in every paired seed on these tasks. With exact target detection, chunked GD beats FEAS-Adam on 6 of the eight, and single-tensor GD on 7. Circles is the task where removing gradient accumulation changes the ordering. The eight-puzzle network still favors minibatching because each full gradient processes 8,000 records.

Task	GD (256)	GD (one batch)	SGD	FEAS-SGD	FEAS-Adam
linear	1 / 1.30	1 / 0.66	12 / 3.67	11 / 5.30	102 / 50.07
diabetes	2 / 1.11	2 / 0.83	7 / 2.27	7 / 3.40	13 / 7.20
cancer-logistic	3 / 1.63	3 / 1.16	6 / 2.11	2 / 1.46	7 / 4.55
wine-softmax	3 / 1.36	3 / 1.22	16 / 5.07	8 / 4.52	34 / 20.34
linear-margin	3 / 4.90	3 / 1.68	30 / 12.58	25 / 16.14	101 / 77.60
logic-gates	23 / 14.70	23 / 9.75	93 / 36.10	76 / 46.54	40 / 26.87
circles	9 / 14.15	9 / 5.28	10 / 3.84	15 / 8.62	19 / 12.91
eight-puzzle	8 / 78.96	8 / 28.33	11 / 5.66	12 / 8.70	10 / 9.49

For method a, let S_a denote setup time, K_a the update count, B_a the examples per update, and $R_a = K_a B_a/(T_a - S_a)$ the realized gradient-example throughput. Then for FEAS (F) and GD (G),

$$T_{\mathrm{F}} - T_{\mathrm{G}} = (S_{\mathrm{F}} - S_{\mathrm{G}}) + \frac{K_{\mathrm{F}} B_{\mathrm{F}}}{R_{\mathrm{F}}} - \frac{K_{\mathrm{G}} B_{\mathrm{G}}}{R_{\mathrm{G}}}.$$

For cancer classification, FEAS-Adam used 224 gradient examples to GD's 1,092, yet took 4.55 versus 1.63 ms. A full gradient needs one parameter update per scan. Each adaptive minibatch needs an indexed gather, an importance-weighted backward pass, a variance observation, and its own parameter update. The draw and gather, proxy, and optimizer update account for about 66% of FEAS-Adam's measured post-setup time on that task and 11% on MNIST CNN. The difference reflects the larger share of time spent evaluating the network on image batches.

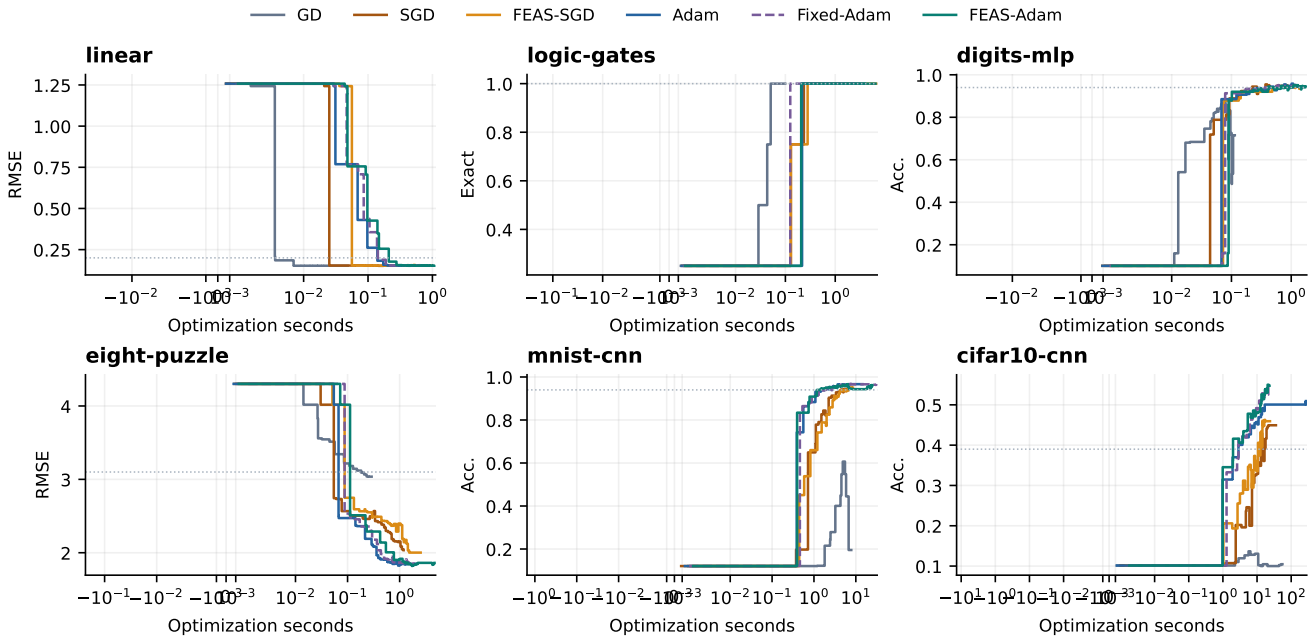


Figure 3: The same validation curves against measured optimization seconds. Small examples can favor the low per-update overhead of the full-batch implementation.

Step size and exact curvature. For the linear task let Z be the training matrix of standardized inputs with an added bias column. With $H = 2Z^\top Z/N$ and the least-squares solution $\theta^\star$, GD has

$$\theta^+ - \theta^\star = (I - \eta H)(\theta - \theta^\star).$$

The empirical eigenvalues of H range from 1.8124 to 2.1477. At $\eta = 0.45$, the spectral contraction factor is at most 0.1845. For the three starts the observed one-step parameter-error ratios lie between 0.0882 and 0.0945. Two starts meet the validation RMSE target of 0.20 in one step and the third in two. Using $\eta = 0.10$, the SGD pilot's choice, full GD takes a median of ten updates. Increasing FEAS-SGD from 0.10 to 0.45 reduces its median update count from eleven to two. The equal-rate comparisons below give updates and milliseconds. At the same rate, FEAS-SGD uses 64 gradient examples versus 1,152 for GD. GD is still faster, especially in one batch.

Control	GD (256)	GD (one batch)	Uniform SGD	FEAS-SGD
linear	1 / 1.30	1 / 0.66	3 / 1.22	2 / 1.57

The first FEAS draw on the linear task has proportional counts and removes only 4.5–5.0% of the uniform minibatch's initial covariance trace. The variance penalty can be computed exactly. If $M_{\mathrm{val}} = Z_{\mathrm{val}}^\top Z_{\mathrm{val}}/N_{\mathrm{val}}$ and Σ is the covariance of an unbiased gradient estimate, then

$$\mathbb{E}[L_{\mathrm{val}}(\theta - \eta\widehat{g})] = L_{\mathrm{val}}(\theta - \eta g) + \eta^2 \operatorname{tr}(M_{\mathrm{val}}\Sigma).$$

At $B = 32$ and $\eta = 0.45$, the expected first-step validation MSE for the stratified estimator ranges from 0.281 to 0.509 across starts. The full-gradient MSE ranges from 0.034 to 0.044. These values follow from the finite gradient matrix without Monte Carlo estimation.

Exact duplicate control. The logic-gate task repeats each of four inputs 128 times with the same targets. The loss over four representatives equals the empirical loss on all 512 records. We checked equality

within each group, used the same SGD learning rate of 1, and compared equal caps of 40 updates. Drawing one item per group gives the full gradient in exact arithmetic. In both four-example arms the paired seeds reach the same update as GD on all 512 rows, or all three arms miss the target for that seed. Entries give gradient examples and milliseconds. A sampler can save work compared with scanning the repeated input. Direct evaluation of the four distinct records uses the same number of gradients and avoids the sampling overhead.

Control	Repeated GD (256)	Repeated GD (one batch)	Four-record GD	FEAS-SGD
logic-gates	11,776 / 13.30	11,776 / 8.80	92 / 7.84	92 / 12.51

CNNs and matched minibatch controls. For the CNN replays, entries give gradient examples and seconds. FEAS-Adam beats uniform Adam on median time in 3 of the three CNNs. It beats fixed stratification in 1 of the three. Fixed and adaptive allocations can save exposures in the same problem. The fixed control separates this effect from the extra variance estimation and allocation refresh.

Task	Adam	Fixed-Adam	FEAS-Adam
mnist-cnn	9,216 / 0.882	6,816 / 0.678	5,664 / 0.619
fashion-cnn	5,376 / 0.572	3,936 / 0.402	3,936 / 0.454
cifar10-cnn	8,640 / 1.341	6,048 / 0.965	5,952 / 1.065

5.3 Calibrating batch size and peak step

Bessa et al. [22] calibrate peak centroid step sizes over a batch-size grid, identify a batch-size knee near the validation loss floor, and fit power laws across data sizes, cluster counts, and epoch budgets. They also decay the centroid step on a cosine schedule. We transfer the calibration procedure and the schedule to neural optimization. The original coefficients describe an exponential moving average of centroid sufficient statistics. Network training has different curvature, gradient variance, and execution costs.

A batch and step law for the descent bound. Suppose the current loss is L_s-smooth. Fix an allocation share q and write $G = \|g\|^2 > 0$, $V = \mathcal{V}_v(q)$. The conditional expected one-update improvement is bounded below by

$$d(B, \eta) = \eta G - \frac{L_s \eta^2}{2}(G + V/B).$$

For each B, this concave quadratic is maximized at

$$\eta_B^\star = \frac{1}{L_s} \frac{B}{B + V/G}.$$

The bound favors a nearly linear step increase at small B relative to V/G and saturates at $1/L_s$ as the batch grows. Model the cost of an update locally as $c_0 + c_1 B$, with positive coefficients and $V > 0$. Maximizing $d(B, \eta_B^\star)/(c_0 + c_1 B)$ is equivalent to minimizing

$$(G + V/B)(c_0 + c_1 B) = G c_0 + V c_1 + G c_1 B + V c_0 / B.$$

The last two terms balance at $B^\star = \sqrt{(c_0/c_1)(V/G)}$. For a continuous interval of feasible batch sizes the optimum is this value clipped to the interval. On an integer interval only its adjacent feasible integers need comparison. At $V = 0$ the smallest feasible batch maximizes the bound per second. This is a local, one-step calculation with frozen q and V. When count rounding or adaptive allocation makes V batch-dependent, one can evaluate the same criterion separately on candidate batch sizes. Neither the affine timing model nor a one-step bound identifies the time to a general validation threshold.

Transfer experiment. We choose seven tasks spanning linear regression, classification, image MLPs, and a learned search heuristic. SGD is the base optimizer for linear and diabetes. Adam is the base for the other five. All variants keep each task's original data split, target, exposure cap C, and fixed stratum definition. Denote the original batch and rate by B_0, η_0. In a validation-only pilot with seeds 913 and 1499, we sweep batches $B_0/2, B_0, 2B_0, 4B_0$, clipped to $[m, N]$, and peak rates $\eta_0/2, \eta_0, 2\eta_0, 4\eta_0$. For the cosine arm the rate after e gradient examples is

$$\eta(e) = \eta_{\text{peak}} \left[0.1 + 0.45(1 + \cos(\pi \min(e/P, 1)))\right], \qquad P = \min(C, \max(N, 96B_0)).$$

Each candidate uses the same pilot cap P. For each schedule and each batch, choose the rate with lowest median terminal validation loss among candidates stable in both pilot seeds. The selected batch is the largest grid point within 2% of that schedule's best pilot loss. This small grid calibrates a knee per task. The proof of conditional unbiasedness still applies since the choice is fixed before deployment and each count vector is chosen before its corresponding batch.

Held-out initializations. The evaluation uses five seeds (13, 37, 71, 101, 151) disjoint from the pilot. We compare the original method, a rate-only arm at B_0, the calibrated constant-step knee, and the calibrated cosine knee. We also replay the cosine arm's batch and peak with a constant rate, replace its sampler with uniform draws, and run one-tensor full GD at its archived pilot rate. All methods are validated after every update to locate the target and replayed without intervening validation for timing. After a warmup we retain three replay times per seed. Pilot time is separate from the time of training with frozen settings. The first table reports gradient examples and milliseconds.

Task	Original	Rate only	Constant knee	Cosine knee
linear	384 / 6.94	768 / 12.67	1,536 / 6.47	768 / 3.52
diabetes	320 / 4.33	608 / 7.89	640 / 2.37	384 / 1.57
cancer-logistic	448 / 10.58	160 / 4.29	1,024 / 5.30	1,024 / 5.55
circles	608 / 13.78	192 / 4.58	176 / 7.30	176 / 7.32
digits-mlp	2,016 / 48.27	2,432 / 53.96	1,584 / 63.93	1,616 / 66.62
eight-puzzle	768 / 6.83	480 / 5.21	288 / 5.42	192 / 3.80
mnist-mlp	11,712 / 169.64	11,712 / 165.92	8,976 / 203.13	8,160 / 196.33

The calibrated cosine arm was faster than the original FEAS configuration on 5 of 7 tasks by median optimization time. Diabetes went from 4.33 to 1.57 ms and eight-puzzle from 6.83 to 3.80 ms. On eight-puzzle, the median gradient-example count dropped from 768 to 192. Both image MLPs used fewer gradient examples but took longer. The pilot took 2.6 to 13.5 seconds per task on this machine. These training-time savings apply when the calibration is reused.

The following controls give milliseconds to the same validation target. A target is marked NR if its median time is censored. The uniform control uses the same batch size, peak, and decay horizon as the adaptive arm.

Task	B	Peak	GD	Uniform	Same peak	FEAS cosine
linear	128	0.2	0.93	2.29	3.45	3.52
diabetes	128	0.1	0.84	0.86	1.59	1.57
cancer-logistic	128	0.048	1.14	3.21	5.52	5.55
circles	16	0.048	5.17	5.77	7.85	7.32
digits-mlp	16	0.024	NR	57.23	101.26	66.62
eight-puzzle	48	0.016	26.09	2.70	3.94	3.80
mnist-mlp	48	0.024	NR	142.49	882.53	196.33

At the same peak rate and batch size, cosine and constant FEAS used the same median gradient-example count on 5 tasks. Uniform sampling was faster on 7 of 7 tasks at the same batch size, peak rate, and schedule.

5.4 Extended full-batch vision runs

Several main vision tasks censored the GD comparator before it reached the validation target. We therefore reran the same full-batch optimizer and its previously selected step size from each paired initialization on all four vision tasks, with four times the main exposure cap and a validation check after every full gradient. The Feas results in the table are from the main study. In a censored GD row the displayed time is a lower bound on its median first observed crossing, obtained by assigning the recorded end-of-run time as a lower bound for each censored seed and taking the median. No numerical speedup ratio is assigned beyond that bound.

Task	GD cap	GD examples	GD s	FEAS examples	FEAS s
mnist-mlp	288,000	$\mathrm{NR}^{0/3}$	> 2.26	12,000	0.525
mnist-cnn	160,000	$\mathrm{NR}^{0/3}$	> 50.02	6,720	1.169
fashion-cnn	192,000	$\mathrm{NR}^{0/3}$	> 33.27	5,376	1.334
cifar10-cnn	252,000	$\mathrm{NR}^{0/3}$	> 87.22	7,008	1.980

The adaptive method crossed the validation target within the measured time window of the extended GD comparator on 4 of the 4 vision tasks. The extension uses the same fixed GD step sizes as the main study. For the CNNs the pilot selected the top step size in the grid, and these prolonged runs remain censored. They show a timing advantage over the tested configuration and leave the outcome of a different long-horizon GD tuning policy open.

5.5 Fixed label budget

The separate label-budget experiment uses a deterministic quarter of the original training labels for Digits, MNIST MLP, and AG News. The validation and test data are unchanged. We increase the allowed passes by four so the gradient-example cap is approximately equal to the full-label experiment. Each method uses the same subset within a task. A new validation-only pilot tunes each base method for this subset. Below are endpoint median test accuracies. The final column gives the full-pool Feas-Adam result for context. Within a row the first four methods have the same label count. These figures compare methods under fixed acquisition budgets, while the earlier thresholds primarily measure optimization exposures.

Task	Labels	GD	Adam	Fixed-Adam	FEAS-Adam	FEAS full pool
digits-mlp	287	0.936	0.939	0.936	0.942	0.961
agnews-mlp	3,000	0.428	0.705	0.698	0.696	0.774
mnist-mlp	3,000	0.761	0.944	0.940	0.918	0.935

5.6 Control evaluations

The FrozenLake evaluation computes the exact goal probability of the fitted greedy policy. The CartPole evaluation applies the fitted imitator on 24 fixed resets for up to 500 steps. A method can fit the static classification targets and still have a different control return, because its own actions alter its subsequent inputs. These diagnostics are reported separately from the validation targets and training budgets.

Evaluation	GD	SGD	FEAS-SGD	Adam	Fixed-Adam	FEAS-Adam
FrozenLake success	0.511	0.013	0.036	0.740	0.740	0.740
CartPole return	258.2	500.0	500.0	500.0	500.0	500.0

6 Reproduction and interpretation

The source archive contains the sampler, loss and model definitions, the complete per-seed traces, pilot learning-rate selection, the control evaluations, and scripts to regenerate the tables and figures. It also includes exhaustive small-instance checks of theorem 2, tests for the gradient weights and variance statistic, and numerical verification of the paired ReLU trajectory. Dataset retrieval checks SHA-256 and the archive carries only code, paper sources, and derived results. The same conditional gradient identity applies to finite datasets whose stratum names are supplied by other metadata.

The comparison with full GD mixes statistical and computational effects. A minibatch can change the parameters before a full data scan would finish. The matching SGD and Adam comparisons ask the narrower question of what changes when only the sampler changes. Fixed-Adam isolates the adaptive entropy-regularized allocation from stratification itself. The speedup theorems isolate two sources of sample savings, reuse of homogeneous strata and removal of between-stratum variation. Neither theorem requires an advantage for the fitted proxy on every task. The variance criterion in theorem 3 specifies the information about the current partition that would justify a variance claim. Timing additionally depends on cost per update, especially for small models.

The paper uses an anonymous byline and was prepared with generative AI assistance for drafting, implementation, and verification. Formal proofs, finite-instance checks, and measured results are identified separately. The implementation records the complete optimization protocol so that each numerical claim can be traced to paired runs.

A All benchmark specifications

The main budgets are gradient-example exposures. Accuracy, exact truth-table success, ARI, and goal success are maximized. RMSE is minimized. FrozenLake success is a probability in the 100-step model. Puzzle RMSE is in moves. Diabetes RMSE uses standardized outcomes.

Task	Train/val/test	Batch	Budget	Measure	Target	Strata
linear	1,152/288/360	32	16,128	RMSE	0.2	4
polynomial	1,152/288/360	32	16,128	RMSE	0.072	4
diabetes	282/71/89	32	7,332	RMSE	0.84	4
logic-gates	512/4/4	16	20,480	Exact	1	4
cancer-logistic	364/91/114	32	8,736	Acc.	0.94	2
iris-softmax	96/24/30	32	4,608	Acc.	0.93	3
wine-softmax	113/29/36	32	3,955	Acc.	0.94	3
two-moons	1,408/352/440	32	36,608	Acc.	0.93	2
circles	1,408/352/440	32	36,608	Acc.	0.93	2
digits-mlp	1,149/288/360	32	21,831	Acc.	0.94	10
digits-pca	1,149/288/360	64	28,725	RMSE	0.186	4
digits-autoencoder	1,149/288/360	64	34,470	RMSE	0.207	4
soft-kmeans	1,152/288/360	48	16,128	ARI	0.9	4
movielens-mf	12,000/2,000/2,500	96	96,000	RMSE	0.99	4
agnews-mlp	12,000/1,500/2,000	96	72,000	Acc.	0.79	4
sequence-gru	2,620/656/820	64	41,920	Acc.	0.9	2
frozenlake-value	16/16/16	16	1,920	Success	0.55	3
cartpole-imitation	2,560/640/800	48	38,400	Acc.	0.96	2
eight-puzzle	8,000/2,000/2,000	96	80,000	RMSE	3.1	4
tic-tac-toe	2,892/724/904	64	52,056	Acc.	0.78	3
linear-margin	1,280/320/400	48	20,480	Acc.	0.94	2
gaussian-mixture	1,152/288/360	48	16,128	ARI	0.9	4
mnist-mlp	12,000/1,500/2,500	96	72,000	Acc.	0.93	10
mnist-cnn	8,000/1,500/2,500	96	40,000	Acc.	0.94	10
fashion-cnn	8,000/1,500/2,500	96	48,000	Acc.	0.78	10
cifar10-cnn	9,000/1,500/2,500	96	63,000	Acc.	0.39	10

B First validation crossing in gradient examples

Each superscript $h/3$ records the number of seeds reaching the target if fewer than all three reached it. NR means fewer than two seeds reached the target. The CSV and JSON results also include index-adjusted exposures and distinct examples used for gradients.

Task	GD	SGD	FEAS-SGD	Adam	Fixed-Adam	FEAS-Adam
linear	$1,152$	896	896	$3,584$	$3,584$	$3,584$
polynomial	$\mathrm{NR}^{1/3}$	896	896	$2,688$	$2,688$	$2,688$
diabetes	564	416	416	416	416	416
logic-gates	$11,776^{2/3}$	$2,288$	$2,288$	$1,152$	$1,152$	$1,152$
cancer-logistic	$1,092$	512	512	992	512	512
iris-softmax	$4,128$	$\mathrm{NR}^{1/3}$	$4,096$	$\mathrm{NR}^{0/3}$	$\mathrm{NR}^{0/3}$	$\mathrm{NR}^{0/3}$
wine-softmax	452	672	448	$1,120$	$1,120$	$1,120$
two-moons	$\mathrm{NR}^{0/3}$	$8,160$	$8,160$	$4,096$	$4,096$	$4,096$
circles	$12,672$	$2,048$	$2,048$	$2,048$	$2,048$	$2,048$
digits-mlp	$\mathrm{NR}^{0/3}$	$6,080$	$7,296$	$7,296$	$6,080$	$4,864$
digits-pca	$\mathrm{NR}^{0/3}$	$\mathrm{NR}^{0/3}$	$\mathrm{NR}^{0/3}$	$6,400$	$8,000$	$8,000$

Task	GD	SGD	FEAS-SGD	Adam	Fixed-Adam	FEAS-Adam
digits-autoencoder	$\mathrm{NR}^{0/3}$	$\mathrm{NR}^{0/3}$	$\mathrm{NR}^{0/3}$	5,760	5,760	5,760
soft-kmeans	0	0	0	0	0	0
movielens-mf	$\mathrm{NR}^{0/3}$	$\mathrm{NR}^{0/3}$	$\mathrm{NR}^{0/3}$	$\mathrm{NR}^{0/3}$	$\mathrm{NR}^{0/3}$	$\mathrm{NR}^{0/3}$
agnews-mlp	$\mathrm{NR}^{0/3}$	$\mathrm{NR}^{0/3}$	$\mathrm{NR}^{0/3}$	$\mathrm{NR}^{1/3}$	$24,000^{2/3}$	$24,000^{2/3}$
sequence-gru	$\mathrm{NR}^{0/3}$	11,648	7,040	2,368	2,368	2,368
frozenlake-value	$\mathrm{NR}^{0/3}$	$\mathrm{NR}^{0/3}$	$\mathrm{NR}^{0/3}$	656	656	864
cartpole-imitation	$\mathrm{NR}^{0/3}$	$\mathrm{NR}^{0/3}$	$\mathrm{NR}^{0/3}$	27,744	29,904	29,904
eight-puzzle	$64,000^{2/3}$	4,512	4,512	4,512	4,512	4,512
tic-tac-toe	$\mathrm{NR}^{0/3}$	43,392	$31,872^{2/3}$	20,288	17,408	14,464
linear-margin	3,840	2,304	2,304	4,560	4,560	5,712
gaussian-mixture	0	0	0	0	0	0
mnist-mlp	$\mathrm{NR}^{0/3}$	40,032	40,032	20,064	16,032	12,000
mnist-cnn	$\mathrm{NR}^{0/3}$	24,480	24,480	11,136	8,928	6,720
fashion-cnn	$\mathrm{NR}^{0/3}$	13,344	13,344	8,064	5,376	5,376
cifar10-cnn	$\mathrm{NR}^{0/3}$	35,040	38,592	10,560	7,008	7,008

C First validation crossing in optimization seconds

These medians use the same first crossings as the exposure table. Brackets indicate a median bounded by the observed times and the censoring deadlines. Validation time is separately recorded in the companion. The main suite uses the original, equal-exposure caps.

Task	GD	SGD	FEAS-SGD	Adam	Fixed-Adam	FEAS-Adam
linear	0.005	0.025	0.056	0.140	0.142	0.210
polynomial	$\mathrm{NR}^{1/3}$	0.030	0.062	0.162	0.128	0.221
diabetes	0.005	0.014	0.023	0.021	0.024	0.032
logic-gates	$0.051^{2/3}$	0.235	0.279	0.216	0.125	0.207
cancer-logistic	0.006	0.014	0.032	0.056	0.026	0.044
iris-softmax	0.048	$\mathrm{NR}^{1/3}$	0.205	$\mathrm{NR}^{0/3}$	$\mathrm{NR}^{0/3}$	$\mathrm{NR}^{0/3}$
wine-softmax	0.004	0.018	0.022	0.046	0.059	0.068
two-moons	$\mathrm{NR}^{0/3}$	0.291	0.453	0.217	0.244	0.290
circles	0.046	0.075	0.131	0.102	0.098	0.145
digits-mlp	$\mathrm{NR}^{0/3}$	0.222	0.434	0.398	0.364	0.409
digits-pca	$\mathrm{NR}^{0/3}$	$\mathrm{NR}^{0/3}$	$\mathrm{NR}^{0/3}$	0.214	0.214	0.267
digits-autoencoder	$\mathrm{NR}^{0/3}$	$\mathrm{NR}^{0/3}$	$\mathrm{NR}^{0/3}$	0.225	0.237	0.332
soft-kmeans	0.001	0.001	0.002	0.001	0.003	0.004
movielens-mf	$\mathrm{NR}^{0/3}$	$\mathrm{NR}^{0/3}$	$\mathrm{NR}^{0/3}$	$\mathrm{NR}^{0/3}$	$\mathrm{NR}^{0/3}$	$\mathrm{NR}^{0/3}$
agnews-mlp	$\mathrm{NR}^{0/3}$	$\mathrm{NR}^{0/3}$	$\mathrm{NR}^{0/3}$	$\mathrm{NR}^{1/3}$	$0.681^{2/3}$	$[2.226, 2.540]^{2/3}$
sequence-gru	$\mathrm{NR}^{0/3}$	1.071	1.088	0.445	0.227	0.269
frozenlake-value	$\mathrm{NR}^{0/3}$	$\mathrm{NR}^{0/3}$	$\mathrm{NR}^{0/3}$	0.066	0.045	0.139
cartpole-imitation	$\mathrm{NR}^{0/3}$	$\mathrm{NR}^{0/3}$	$\mathrm{NR}^{0/3}$	0.407	0.589	0.781
eight-puzzle	$0.183^{2/3}$	0.054	0.088	0.067	0.088	0.112
tic-tac-toe	$\mathrm{NR}^{0/3}$	0.424	$0.587^{2/3}$	0.273	0.266	0.289
linear-margin	0.007	0.032	0.050	0.096	0.092	0.178
gaussian-mixture	0.001	0.001	0.002	0.001	0.001	0.001
mnist-mlp	$\mathrm{NR}^{0/3}$	0.792	1.023	0.545	0.452	0.525
mnist-cnn	$\mathrm{NR}^{0/3}$	4.347	5.309	1.803	1.771	1.169
fashion-cnn	$\mathrm{NR}^{0/3}$	2.888	3.142	1.901	0.902	1.334
cifar10-cnn	$\mathrm{NR}^{0/3}$	13.08	10.67	2.862	2.848	1.980

D Endpoint test score

The final column is the adaptive Feas sampler coupled to Adam. The rest of the columns provide the full GD, uniform, and fixed-stratification controls at the stated cap. Truth-table and FrozenLake entries evaluate the complete finite problem. The other rows use held-out records.

Task	GD	SGD	FEAS-SGD	Adam	Fixed-Adam	FEAS-Adam
linear	0.152	0.153	0.153	0.153	0.153	0.153
polynomial	0.087	0.052	0.055	0.052	0.052	0.053
diabetes	0.718	0.717	0.716	0.722	0.715	0.718
logic-gates	1.000	1.000	1.000	1.000	1.000	1.000
cancer-logistic	0.974	0.974	0.982	0.974	0.974	0.974
iris-softmax	0.900	0.900	0.900	0.833	0.867	0.867
wine-softmax	1.000	1.000	1.000	1.000	1.000	1.000
two-moons	0.907	0.964	0.973	0.968	0.973	0.970
circles	0.984	0.998	0.998	0.998	0.998	0.998
digits-mlp	0.714	0.956	0.956	0.958	0.967	0.961
digits-pca	0.265	0.243	0.243	0.159	0.159	0.159
digits-autoencoder	0.364	0.272	0.272	0.132	0.132	0.133
soft-kmeans	1.000	1.000	1.000	1.000	1.000	1.000
movielens-mf	1.149	1.036	1.037	1.123	1.134	1.129
agnews-mlp	0.333	0.758	0.757	0.776	0.772	0.774
sequence-gru	0.802	1.000	1.000	1.000	1.000	1.000
frozenlake-value	0.511	0.013	0.036	0.740	0.740	0.740
cartpole-imitation	0.894	0.920	0.919	0.973	0.973	0.971
eight-puzzle	3.217	2.064	2.026	1.856	1.862	1.846
tic-tac-toe	0.628	0.768	0.765	0.861	0.862	0.848
linear-margin	0.985	0.980	0.983	0.983	0.978	0.978
gaussian-mixture	1.000	1.000	1.000	1.000	1.000	1.000
mnist-mlp	0.308	0.945	0.941	0.951	0.955	0.935
mnist-cnn	0.205	0.956	0.959	0.976	0.973	0.972
fashion-cnn	0.176	0.828	0.832	0.852	0.853	0.860
cifar10-cnn	0.107	0.456	0.455	0.485	0.538	0.531

References

[1] Anonymous. *Replication and Persistence in Spacetime*. Manuscript submitted to Higher Categorical Semantics, 2026. Chapters 10 and 12.

[2] Jerzy Neyman. On the two different aspects of the representative method. *Journal of the Royal Statistical Society*, 97(4), 1934. DOI.

[3] Deanna Needell, Nathan Srebro, and Rachel Ward. Stochastic gradient descent, weighted sampling, and the randomized Kaczmarz algorithm. *NeurIPS*, 2014. Proceedings.

[4] Peilin Zhao and Tong Zhang. Stochastic optimization with importance sampling for regularized loss minimization. *ICML*, PMLR 37, 2015. Proceedings.

[5] Guillaume Bouchard, Théo Trouillon, Julien Perez, and Adrien Gaidon. Online learning to sample. 2015. arXiv:1506.09016.

[6] Angelos Katharopoulos and François Fleuret. Not all samples are created equal. *ICML*, PMLR 80, 2018. Proceedings.

[7] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. *NeurIPS*, 2013. Proceedings.

[8] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2015. arXiv:1412.6980.

[9] Massachusetts Institute of Technology. 6.036 Introduction to Machine Learning, course description. Course site.

[10] Stanford University. CS229 Machine Learning, syllabus and lecture materials. Course site.

[11] Stanford University. CS221 Artificial Intelligence: Principles and Techniques, spring 2026. Course site.

[12] University of California, Berkeley. CS 188 Introduction to Artificial Intelligence, catalog. Course site.

[13] Scikit-learn contributors. Toy datasets documentation. Documentation.

[14] Pierre Baldi and Kurt Hornik. Neural networks and principal component analysis. *Neural Networks*, 2(1):53–58, 1989. Author-hosted paper.

[15] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. Publisher.

[16] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST. 2017. arXiv:1708.07747.

[17] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. CIFAR-10 and CIFAR-100 datasets. Dataset page.

[18] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. *NeurIPS*, 2015. arXiv.

[19] GroupLens Research. MovieLens 100K dataset and standard splits. Dataset page.

[20] Farama Foundation. FrozenLake, Gymnasium v1.2.0 documentation. Environment specification.

[21] Farama Foundation. CartPole, Gymnasium documentation. Environment specification.

[22] Swann Bessa, Pierre-Emmanuel Mazaré, Gergely Szilvasy, Hugo Touvron, David Picard, Matthijs Douze, and Hervé Jégou. e-means: outperforming k-means (in one-tenth the epochs). 2026. arXiv:2611.09501.