GHOSTMATCH: VALIDATION-FREE ONLINE BATCH SELECTION FOR LLM TRAINING VIA SUBMODULAR GRADIENT MATCHING

Anonymous authors
Paper under double-blind review

ABSTRACT

Online batch selection reduces LLM training cost by replacing a candidate batch with a smaller informative subset. Existing selection methods often rely on held-out validation data, while their objectives may lack approximation guarantees. We introduce GhostMatch, a validation-free method that selects a weighted subset to preserve the gradient geometry of the current candidate batch. Casting this problem as facility location in gradient space yields a monotone submodular objective, so greedy selection achieves the classical $(1 - 1/e)$ approximation guarantee. We also provide a nonconvex convergence analysis that directly relates the optimization error from subset selection to its gradient representation error. We use ghost inner products to compute the exact pairwise gradient geometry needed for facility-location selection without materializing per-example gradients. Across four LLM fine-tuning configurations from 1.1B to 8B parameters, GhostMatch remains competitive with validation-guided methods while using no held-out data and training on only half of each candidate batch, including outperforming the validation-guided OPUS baseline on Mistral-7B (0.609 vs. 0.572 F1). A controlled ablation further shows that exact gradient geometry consistently outperforms an approximate alternative. Code is available at `https://anonymous.4open.science/r/ghostmatch-review-E1C7/`.

1 INTRODUCTION

Fine-tuning large language models (LLMs) on instruction-following and task-specific data is computationally expensive, and not every example in a training batch contributes equally to a given optimization step. Online batch selection reduces this cost by scoring a larger candidate batch at each step and training only on a smaller, more informative subset, improving convergence and downstream performance relative to training on randomly sampled batches of the same size (Loshchilov & Hutter, 2015; Katharopoulos & Fleuret, 2018). Because evaluating every possible subset of a candidate batch is combinatorially intractable, selection methods in this line of work uniformly rely on greedy subset construction, which is efficient but is only as reliable as the combinatorial structure of the objective it greedily maximizes. When that objective is monotone submodular, greedy selection carries a classical worst-case approximation guarantee (Nemhauser et al., 1978); without such structure, it is an uncertified heuristic whose empirical performance on one set of batches provides no guarantee for another.

The strongest recent methods in this space, GREATS (Wang et al., 2024) and OPUS (Wang et al., 2026), score candidates by their estimated effect on a held-out validation set, using ghost inner products to make this tractable at LLM scale. Both are effective in practice, but both depend on a validation set that already reflects the target task, an assumption that does not always hold, for instance in continual fine-tuning or multi-task settings where the evaluation criteria are not fixed in advance. Both methods are also greedy over a Taylor-approximated validation-utility score; in Appendix D, we show this score is not submodular in general, so neither method's greedy selection carries an approximation guarantee.

We introduce GhostMatch, an online batch selection method that removes both limitations. Rather than scoring candidates against an external validation target, GhostMatch selects a weighted subset of the candidate batch that best reconstructs the batch's own gradient, posed as a facility-location problem over candidate gradients and computed using the same ghost inner-product machinery as

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

GREATS, without requiring validation data or advance knowledge of the downstream task (Sections 4–5). Facility location is monotone submodular, so greedy selection for GhostMatch retains the classical $(1 - 1/e)$ approximation guarantee. We complement this combinatorial guarantee with a nonconvex stationarity bound (Section 6) that relates the optimization error of training on the selected subset to the quality of its gradient representation, recovering standard mini-batch SGD convergence guarantees as this representation error vanishes.

We evaluate GhostMatch across multiple fine-tuning configurations spanning several model scales and task types, comparing it against validation-directed selection methods and against standard training on the full candidate batch. Validation-directed methods have access to a held-out sample from the target domain, information GhostMatch does not use, so it is unsurprising that they are often the strongest performers, and our results confirm this. The finding worth reporting is what happens without that advantage. GhostMatch, using no validation data, remains competitive with the strongest validation-directed methods, and despite training on only half of each candidate batch, often closely tracks or exceeds training on the full batch that its selected gradient is designed to approximate.

Contributions.

- **GhostMatch**, a validation-free online batch selection method that formulates representative subset selection as facility location over ghost-computed candidate-batch gradients (Section 5).
- A **submodularity guarantee for GhostMatch**. Because its facility-location objective is monotone submodular, greedy selection inherits the classical $(1 - 1/e)$ approximation guarantee (Nemhauser et al., 1978). We show this guarantee does *not* extend to GREATS' Taylor-approximated validation-utility score (Proposition 3, Appendix D), a gap that applies equally to OPUS, despite both methods also relying on greedy selection.
- A **nonconvex convergence bound** (Theorem 1) relating the optimization error of training on a selected subset to that subset's gradient-representation error, specialized for GhostMatch in Corollary 1, which recovers standard mini-batch SGD convergence rates as representation error vanishes.
- A **linear convergence guarantee under the Polyak–Łojasiewicz condition** (Theorem 2, Appendix C.1), showing that GhostMatch's expected objective suboptimality contracts linearly to a neighborhood whose size is controlled explicitly by mini-batch sampling variance and subset representation error.
- An **empirical evaluation** across four model-task configurations showing that GhostMatch remains competitive with validation-directed baselines despite using no validation data; a controlled ablation showing that GhostMatch's exact gradient geometry consistently improves downstream performance relative to its zeroth-order counterpart at the main training horizons when the selection objective and rule are held fixed; and a characterization of a cold-start setting in which full-batch training is initially strongest, with extended-horizon experiments showing that this advantage is transient (Section 7 and Appendix F).

2 Related Work

Classical coreset methods such as CRAIG (Mirzasoleiman et al., 2020) and GRAD-MATCH (Killamsetty et al., 2021) select weighted subsets whose aggregate gradient approximates that of a larger training set, with CRAIG framing this as a facility-location problem, the same structure GhostMatch adopts. Both target offline or epoch-level selection and do not address the per-step, LLM-scale setting considered here, where materializing per-example gradients for a full training set is infeasible.

The dominant approach to online LLM batch selection is validation-directed. GREATS (Wang et al., 2024) scores candidates by their estimated effect on a held-out validation loss, computed efficiently with ghost inner products. OPUS (Wang et al., 2026), developed for LLM pretraining, extends this with a Gram-matrix redundancy penalty and an optimizer-aware preconditioning term. Both are effective given a representative validation set, but both select greedily over a second-order Taylor approximation of validation utility. We show in Appendix D that this approximation is not monotone submodular for GREATS, and expect the same gap for OPUS's analogous term, so neither method's greedy selection carries an approximation guarantee.

CoLM (Nguyen et al., 2025) is the closest validation-free alternative. It performs selection separately within each sufficiently large source in the data mixture. A source can correspond to a data

group such as an instruction category; when source labels are unavailable, CoLM instead constructs sources by clustering model hidden states. Examples from small sources are retained in full. CoLM also selects over a sparsified, zeroth-order gradient estimate rather than an exact one. GhostMatch applies one selection directly to the full batch with no source labels, using exact ghost gradients throughout, a difference our ablation in Section 7.3 suggests matters for CoLM as well.

GhostMatch is, to our knowledge, the first method to combine validation-free online LLM selection, ghost-computed gradient geometry, and a monotone submodular selection objective. This structure gives greedy selection the classical $(1 - 1/e)$ approximation guarantee. We further provide a nonconvex stationarity bound that ties the optimization error induced by subset selection to the representation cost realized by the selected subset, recovering the standard mini-batch SGD bound as that cost vanishes.

Notation. Sans-serif capital letters, such as S, denote finite sets, and $|\mathsf{S}|$ denotes the cardinality of S. For a positive integer n, we write $[n] = \{1, \dots, n\}$. Bold capital letters, such as $\mathbf{G}$, denote matrices, and bold lowercase letters, such as $\mathbf{w}$, denote vectors. Scalar quantities are written in ordinary lowercase, such as η, k, and γ. $\lceil x \rceil$ denotes the ceiling of the real number x. For a vector $\mathbf{x}$, $\|\mathbf{x}\|$ denotes its Euclidean norm, and $\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^\top \mathbf{y}$ denotes the standard inner product.

3 Problem Formulation

We study online subset selection during the training of a large language model, including the fine-tuning setting evaluated in this paper. Let $\mathsf{D} = \{z_i\}_{i=1}^n$ denote the training set and let $\ell(\mathbf{w}, z)$ denote the loss of parameters $\mathbf{w}$ on example z, giving the empirical risk $f(\mathbf{w}) = \frac{1}{n} \sum_{z \in \mathsf{D}} \ell(\mathbf{w}, z)$. Standard mini-batch training estimates $\nabla f(\mathbf{w})$ using a randomly sampled batch, updating parameters in the direction of that batch's mean gradient. Our goal is to obtain an equally good gradient estimate from a smaller, more carefully chosen set of examples, reducing the number of forward and backward passes required per optimization step.

At iteration t, a candidate batch $\mathsf{B}_t \subseteq \mathsf{D}$ of size b is sampled, and each example $z_i \in \mathsf{B}_t$ has gradient $\mathbf{g}_{t,i} = \nabla_{\mathbf{w}} \ell(\mathbf{w}_t, z_i)$ at the current parameters $\mathbf{w}_t$. Training on the full candidate batch would use the mean gradient $\overline{\mathbf{g}}_t = \frac{1}{b} \sum_{z_i \in \mathsf{B}_t} \mathbf{g}_{t,i}$, which we treat as the reference update direction that a smaller, selected subset should reproduce. The candidate batch thus serves as the per-iteration stochastic approximation to the full training set; performing selection directly over D at every update would require considering all n training examples and would defeat the purpose of scalable online selection.

To reproduce $\overline{\mathbf{g}}_t$ from fewer examples, we select a subset $\mathsf{S}_t \subseteq \mathsf{B}_t$ with $|\mathsf{S}_t| = k < b$ and assign each selected example $z_j \in \mathsf{S}_t$ a nonnegative weight $\gamma_{t,j}$. A single selected example generally cannot stand in for several unselected ones without being reweighted to compensate for the examples it represents, so we require the weights to satisfy $\sum_{z_j \in \mathsf{S}_t} \gamma_{t,j} = b$. This normalization ensures the weighted subset gradient $\widehat{\mathbf{g}}_t = \frac{1}{b} \sum_{z_j \in \mathsf{S}_t} \gamma_{t,j} \mathbf{g}_{t,j}$ is directly comparable to $\overline{\mathbf{g}}_t$, rather than merely an average over a smaller sample.

Following gradient-coreset and gradient-matching methods (Mirzasoleiman et al., 2020; Killamsetty et al., 2021), we formalize subset selection as choosing the weighted subset whose gradient best approximates that of the full candidate batch,

$$\min_{\substack{\mathsf{S} \subseteq \mathsf{B}_t,\ |\mathsf{S}| = k \\ \gamma_j \geq 0,\ \sum_{z_j \in \mathsf{S}} \gamma_j = b}} \left\| \overline{\mathbf{g}}_t - \frac{1}{b} \sum_{z_j \in \mathsf{S}} \gamma_j \mathbf{g}_{t,j} \right\|. \tag{1}$$

Problem equation 1 is a joint combinatorial and continuous optimization over which examples to select and how to weight them, and solving it exactly requires searching over all $\binom{b}{k}$ subsets. Section 5 shows that a particular choice of weighting rule reduces this joint problem to a purely combinatorial one with a tractable and provably good solution.

4 Efficient Gradient Geometry via Ghost Inner Products

The gradient-matching problem in equation 1 is written in terms of the per-example gradients of the candidate batch. For modern LLMs, explicitly constructing and storing these model-sized gradient vectors would make online selection prohibitively expensive. Instead, we represent the candidate gradients through their pairwise inner products, collected in a compact $b \times b$ Gram matrix. As

shown below, this matrix determines the norms and pairwise distances between candidate gradients, which are the quantities needed by the selection construction developed in the next section. We then show how ghost inner products compute this Gram matrix without materializing the underlying per-example gradients.

Let p denote the number of trainable parameters and collect the per-example gradients of the candidate batch into $\mathbf{G}_t \in \mathbb{R}^{b\times p}$, whose ith row is $\mathbf{g}_{t,i}^\top$. We define $\mathbf{K}_t = \mathbf{G}_t\mathbf{G}_t^\top \in \mathbb{R}^{b\times b}$, with entries

$$(\mathbf{K}_t)_{ij} = \langle \mathbf{g}_{t,i}, \mathbf{g}_{t,j}\rangle, \qquad \|\mathbf{g}_{t,i} - \mathbf{g}_{t,j}\|^2 = (\mathbf{K}_t)_{ii} + (\mathbf{K}_t)_{jj} - 2(\mathbf{K}_t)_{ij}.$$

The pairwise Euclidean geometry of the candidate gradients is fully determined by this $b \times b$ matrix $\mathbf{K}_t$, without requiring access to the p-dimensional gradient vectors themselves. It remains to show that $\mathbf{K}_t$ can be computed without first materializing those gradients, which we do next.

We compute $\mathbf{K}_t$ using the ghost inner-product technique developed for LLM training in GREATS (Wang et al., 2024), which follows directly from the structure of backpropagation through a linear layer. Consider a linear layer $\mathbf{y} = \mathbf{W}\mathbf{x}$, and let $\boldsymbol{\theta}_i = \nabla_{\mathbf{y}_i}\ell_i$ denote the backpropagated signal associated with example i, so that its gradient with respect to $\mathbf{W}$ is the outer product $\nabla_{\mathbf{W}}\ell_i = \boldsymbol{\theta}_i\mathbf{x}_i^\top$. The inner product between the parameter gradients of two examples then factorizes as

$$\langle \nabla_{\mathbf{W}}\ell_i, \nabla_{\mathbf{W}}\ell_j\rangle_F = \langle \boldsymbol{\theta}_i, \boldsymbol{\theta}_j\rangle\langle \mathbf{x}_i, \mathbf{x}_j\rangle, \tag{2}$$

where $\langle \mathbf{A}, \mathbf{B}\rangle_F = \mathrm{tr}(\mathbf{A}^\top\mathbf{B})$ denotes the Frobenius inner product. Both $\mathbf{x}_i$ and $\boldsymbol{\theta}_i$ arise naturally during the forward and backward passes and can be accessed without ever constructing the parameter gradient itself, so equation 2 gives the inner product between two per-example gradients without materializing either one. This factorization applies to any layer whose trainable parameters enter linearly, which covers every trainable weight matrix in a standard transformer, including all attention and MLP projections. Nonlinear operations such as softmax attention and pointwise activations have no trainable parameters of their own and therefore require no ghost decomposition; the only nonlinearity with trainable parameters, LayerNorm's elementwise gain and bias, enters linearly in those parameters and admits the same treatment.

For a model with multiple trainable layers, the inner product between two full parameter gradients is the sum of the corresponding layerwise inner products. Our experiments train with LoRA parameters, for which we use the LoRA-specific ghost factorization of GREATS (Wang et al., 2024). We therefore obtain the candidate-batch gradient geometry $\mathbf{K}_t$ directly, without ever materializing model-sized per-example gradient vectors. Once $\mathbf{K}_t$ is available, every similarity and distance needed for gradient matching is evaluated entirely in this $b \times b$ batch geometry, so the high-dimensional parameter space enters selection only implicitly, through the ghost-computed Gram matrix.

5 Submodular Gradient Matching

5.1 From Gradient Matching to Representative Selection

Problem equation 1 couples two decisions, which examples to select and how to weight them. We remove the second decision by defining the weights directly from the selected subset. The candidate batch B_t contains b examples, from which we select a smaller subset S of k representatives. Given any such subset, each example in the candidate batch is assigned to the selected representative whose gradient is closest to its own. Each selected representative then receives a weight equal to the number of candidate examples assigned to it. Thus, once the subset is chosen, its weights are determined automatically. As shown below, this construction yields a direct upper bound on the gradient-matching error.

For clarity, write the candidate batch as $\mathsf{B}_t = \{z_1, \ldots, z_b\}$. Given a subset $\mathsf{S} \subseteq \mathsf{B}_t$, assign each candidate z_i to the selected example in S whose gradient is closest to its own, $\sigma_{\mathsf{S}}(i) \in \arg\min_{j:\, z_j\in\mathsf{S}} \|\mathbf{g}_{t,i} - \mathbf{g}_{t,j}\|$. For each selected example $z_j \in \mathsf{S}$, let $\mathsf{C}_j := \{i \in [b] : \sigma_{\mathsf{S}}(i) = j\}$ denote the candidates assigned to it, and set $\gamma_j := |\mathsf{C}_j|$. Because the sets $\{\mathsf{C}_j : z_j \in \mathsf{S}\}$ partition $[b]$, these weights satisfy $\sum_{z_j\in\mathsf{S}} \gamma_j = b$ and are therefore feasible for equation 1.

With this choice of weights, $\widehat{\mathbf{g}}_t = \frac{1}{b}\sum_{i=1}^b \mathbf{g}_{t,\sigma_{\mathsf{S}}(i)}$, so

$$\|\overline{\mathbf{g}}_t - \widehat{\mathbf{g}}_t\| = \left\| \frac{1}{b}\sum_{i=1}^{b} \big(\mathbf{g}_{t,i} - \mathbf{g}_{t,\sigma_{\mathsf{S}}(i)}\big) \right\| \le \frac{1}{b}\sum_{i=1}^{b} \min_{j:\, z_j\in\mathsf{S}} \|\mathbf{g}_{t,i} - \mathbf{g}_{t,j}\|, \tag{3}$$

where the inequality follows from the triangle inequality.

This motivates the representation cost $c_t(\mathsf{S}) := \sum_{i=1}^{b} \min_{j:\, z_j \in \mathsf{S}} \|\mathbf{g}_{t,i} - \mathbf{g}_{t,j}\|$, so that equation 3 gives $\|\overline{\mathbf{g}}_t - \widehat{\mathbf{g}}_t\| \leq c_t(\mathsf{S})/b$. Selecting a subset with small representation cost therefore directly controls the discrepancy between the weighted subset gradient and the candidate-batch gradient, and every pairwise distance appearing in $c_t(\mathsf{S})$ is available from the ghost-computed Gram matrix $\mathbf{K}_t$ introduced in Section 4.

5.2 Facility-Location Reformulation

We would therefore like to choose a subset of size k that minimizes $c_t(\mathsf{S})$. This problem can be expressed in a form with a useful combinatorial structure. Let $d_{t,ij} := \|\mathbf{g}_{t,i} - \mathbf{g}_{t,j}\|$ and $m_t := \max_{i,j \in [b]} d_{t,ij}$. We convert distances to nonnegative similarities by setting $s_{t,ij} := m_t - d_{t,ij}$ and define

$$F_t(\mathsf{S}) := \sum_{i=1}^{b} \max_{j:\, z_j \in \mathsf{S}} s_{t,ij}, \qquad F_t(\emptyset) := 0. \tag{4}$$

The function F_t is the classical facility-location objective, where each candidate contributes according to its similarity to the most similar selected example. For every nonempty S, $F_t(\mathsf{S}) = b\, m_t - c_t(\mathsf{S})$, so maximizing F_t is exactly equivalent to minimizing the representation cost $c_t(\mathsf{S})$.

Proposition 1 *Let S_{G} be the size-k subset returned by greedy maximization of equation 4, and let $\mathsf{S}^\star$ be an optimal size-k subset. Then $F_t(\mathsf{S}_{\mathrm{G}}) \geq (1 - 1/e) F_t(\mathsf{S}^\star)$.*

This follows directly because the facility-location objective in equation 4 is monotone submodular (Cornuejols et al., 1977), and greedy maximization of a monotone submodular function under a cardinality constraint achieves the classical $(1 - 1/e)$ approximation guarantee (Nemhauser et al., 1978). Together with equation 3, this provides an efficient greedy selection procedure with a formal approximation guarantee on the facility-location objective; the corresponding bound on representation cost is given in Appendix C.3.

5.3 Greedy Selection and Weighted Update

We now use the submodular objective in equation 4 to construct the subset at iteration t. Starting from $\mathsf{S}^{(0)} = \emptyset$, we iteratively add the candidate with the largest marginal increase in F_t,

$$z_{j^\star} \in \arg \max_{z_j \in \mathsf{B}_t \setminus \mathsf{S}^{(r)}} \left[F_t\Big(\mathsf{S}^{(r)} \cup \{z_j\}\Big) - F_t\Big(\mathsf{S}^{(r)}\Big) \right],$$

and set $\mathsf{S}^{(r+1)} = \mathsf{S}^{(r)} \cup \{z_{j^\star}\}$. After k iterations, the selected subset is $\mathsf{S}_t := \mathsf{S}^{(k)}$. All marginal gains are evaluated from the pairwise similarities already contained in the ghost-computed Gram matrix $\mathbf{K}_t$, so no per-example parameter gradients need to be materialized for selection.

Once S_t is fixed, we recover the representative assignment defined in Section 5.1. Each candidate example is assigned to its nearest selected gradient, and each selected example $z_j \in \mathsf{S}_t$ receives weight $\gamma_{t,j}$ equal to the number of candidates assigned to it. The resulting weighted gradient is $\widehat{\mathbf{g}}_t = \frac{1}{b} \sum_{z_j \in \mathsf{S}_t} \gamma_{t,j} \mathbf{g}_{t,j}$, which approximates the candidate-batch gradient according to equation 3.

The weighted update can be implemented without storing the selected per-example gradients. In practice, we reweight the losses of the selected examples by $\gamma_{t,j}$ and obtain $\widehat{\mathbf{g}}_t$ through an ordinary backward pass on S_t. For SGD, this yields the update $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \widehat{\mathbf{g}}_t$; for adaptive optimizers, $\widehat{\mathbf{g}}_t$ is supplied to the optimizer in place of the candidate-batch gradient. Algorithm 1 in Appendix A gives the complete procedure.

For larger candidate batches, the greedy stage can be further accelerated using stochastic greedy selection while retaining a near-optimal approximation guarantee (Mirzasoleiman et al., 2015). The corresponding stochastic-greedy variant of GhostMatch is given in Algorithm 2 in Appendix A.

6 Convergence Analysis

Section 5 gives GhostMatch a guarantee about selection, not about training. Greedy facility-location selection keeps $c_t(\mathsf{S}_t)$ close to the smallest cost any size-k subset could achieve, but nothing said so far tells us what happens once we actually train on that subset. We answer that question here.

The answer turns out to be simple: the same quantity that bounds how well GhostMatch represents a batch also bounds how much optimization suffers for using that representation instead of the batch itself.

Let $\mathbb{E}_t[\cdot]$ denote expectation given everything known up through iteration t, so $\mathbf{w}_t$ is fixed under $\mathbb{E}_t$. We work under two standard conditions from the theory of stochastic nonconvex optimization (Ghadimi & Lan, 2013; Bottou et al., 2018).

Assumption 1 (Smooth and lower-bounded objective) *f is L-smooth and bounded below by $f_\star$. For all $\mathbf{u}, \mathbf{v}$, $f(\mathbf{u}) \le f(\mathbf{v}) + \langle \nabla f(\mathbf{v}), \mathbf{u} - \mathbf{v} \rangle + L/2\|\mathbf{u} - \mathbf{v}\|^2$, and $f(\mathbf{w}) \ge f_\star$ for all $\mathbf{w}$.*

Assumption 2 (Unbiased candidate-batch gradient) *There exists $\sigma^2 \ge 0$ such that the candidate-batch gradient $\overline{\mathbf{g}}_t := \frac{1}{b} \sum_{z_i \in \mathsf{B}_t} \mathbf{g}_{t,i}$ satisfies $\mathbb{E}_t[\overline{\mathbf{g}}_t] = \nabla f(\mathbf{w}_t)$ and $\mathbb{E}_t\big[\|\overline{\mathbf{g}}_t - \nabla f(\mathbf{w}_t)\|^2\big] \le \sigma^2/b$.*

GhostMatch does not train on $\overline{\mathbf{g}}_t$. It trains on the weighted subset gradient $\widehat{\mathbf{g}}_t$ defined in Section 3, and the gap between the two is what we call the selection error, $\mathbf{e}_t := \widehat{\mathbf{g}}_t - \overline{\mathbf{g}}_t$. Writing the update gradient out in full,

$$\widehat{\mathbf{g}}_t = \underbrace{\nabla f(\mathbf{w}_t)}_{\text{true gradient}} + \underbrace{(\overline{\mathbf{g}}_t - \nabla f(\mathbf{w}_t))}_{\text{sampling error}} + \underbrace{\mathbf{e}_t}_{\text{selection error}},$$

makes visible exactly what GhostMatch adds on top of ordinary mini-batch noise. We already know how large this addition can get. By equation 3, GhostMatch's representative weighting scheme guarantees

$$\|\mathbf{e}_t\| \le \frac{c_t(\mathsf{S}_t)}{b} =: \rho_t. \tag{5}$$

Nothing here assumes $\mathbf{e}_t$ is zero-mean or independent of the sampled batch. Neither property holds in general, since the selected subset is constructed from B_t itself. Under exact greedy selection, this dependence is deterministic, while stochastic greedy introduces additional internal randomness.

6.1 From a Single Step to a Convergence Rate

Before working out what ρ_t buys GhostMatch specifically, it helps to ask a simpler question first. Suppose we run SGD on any gradient of the form $\widehat{\mathbf{g}}_t = \overline{\mathbf{g}}_t + \mathbf{e}_t$,

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \widehat{\mathbf{g}}_t, \qquad \widehat{\mathbf{g}}_t = \overline{\mathbf{g}}_t + \mathbf{e}_t, \tag{6}$$

with no further assumption on where $\mathbf{e}_t$ comes from. How much can this cost us? Proofs for this subsection are given in Appendix B.

Lemma 1 (One-step descent) *Suppose Assumptions 1 and 2 hold. If $\eta_t \le 1/(4L)$,*

$$\mathbb{E}_t[f(\mathbf{w}_{t+1})] \le f(\mathbf{w}_t) - \frac{\eta_t}{2}\|\nabla f(\mathbf{w}_t)\|^2 + \frac{L\eta_t^2\sigma^2}{b} + \frac{5\eta_t}{4}\mathbb{E}_t[\|\mathbf{e}_t\|^2]. \tag{7}$$

One step still decreases the loss the way exact gradient descent would, minus two things. One is the usual cost of sampling, and would be there regardless of how the batch is used. The other belongs entirely to $\mathbf{e}_t$. Summing this bound across the run turns a single-step statement into a statement about the whole trajectory.

Theorem 1 (Convergence with approximate gradients) *Suppose Assumptions 1 and 2 hold, and let $\{\mathbf{w}_t\}_{t=0}^{T}$ evolve according to equation 6 with deterministic step sizes $0 < \eta_t \le 1/(4L)$. Let $\Delta_f := f(\mathbf{w}_0) - f_\star$. Then*

$$\frac{\sum_{t=0}^{T-1} \eta_t \,\mathbb{E}\|\nabla f(\mathbf{w}_t)\|^2}{\sum_{t=0}^{T-1} \eta_t} \le \frac{2\Delta_f}{\sum_{t=0}^{T-1} \eta_t} + \frac{2L\sigma^2}{b}\frac{\sum_{t=0}^{T-1} \eta_t^2}{\sum_{t=0}^{T-1} \eta_t} + \frac{5}{2}\frac{\sum_{t=0}^{T-1} \eta_t \,\mathbb{E}\|\mathbf{e}_t\|^2}{\sum_{t=0}^{T-1} \eta_t}. \tag{8}$$

A small left-hand side is what convergence to a stationary point looks like. The bound says this quantity stays small as long as three things stay small, an initialization term that fades on its own, the usual sampling variance, and the running average of $\mathbb{E}\|\mathbf{e}_t\|^2$. This final term is precisely the component controlled by GhostMatch.

6.2 CONVERGENCE OF GHOSTMATCH

Plugging $\|\mathbf{e}_t\|^2 \le c_t(\mathsf{S}_t)^2/b^2$, from equation 5, into Theorem 1 expresses the selection-error term directly in terms of GhostMatch's representation cost.

Corollary 1 (Convergence of GhostMatch) *Suppose Assumptions 1 and 2 hold, and let* $\{\mathbf{w}_t\}_{t=0}^{T}$ *be generated by GhostMatch with deterministic step sizes* $0 < \eta_t \le 1/(4L)$*. Then*

$$\frac{\sum_{t=0}^{T-1} \eta_t \,\mathbb{E}\|\nabla f(\mathbf{w}_t)\|^2}{\sum_{t=0}^{T-1} \eta_t} \le \frac{2\Delta_f}{\sum_{t=0}^{T-1} \eta_t} + \frac{2L\sigma^2}{b}\frac{\sum_{t=0}^{T-1} \eta_t^2}{\sum_{t=0}^{T-1} \eta_t} + \frac{5}{2b^2}\frac{\sum_{t=0}^{T-1} \eta_t \,\mathbb{E}\big[c_t(\mathsf{S}_t)^2\big]}{\sum_{t=0}^{T-1} \eta_t}. \tag{9}$$

The proof is a direct substitution, given in Appendix B. To see what it says in practice, fix a constant step size $\eta_t = \eta \le 1/(4L)$. The corollary becomes

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\|\nabla f(\mathbf{w}_t)\|^2 \le \frac{2\Delta_f}{\eta T} + \frac{2L\eta\sigma^2}{b} + \frac{5}{2b^2 T}\sum_{t=0}^{T-1} \mathbb{E}\big[c_t(\mathsf{S}_t)^2\big],$$

and writing $\delta_T^2 := \frac{1}{b^2 T}\sum_{t=0}^{T-1} \mathbb{E}\big[c_t(\mathsf{S}_t)^2\big]$ for the run's average normalized representation error turns this into

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\|\nabla f(\mathbf{w}_t)\|^2 \le \frac{2\Delta_f}{\eta T} + \frac{2L\eta\sigma^2}{b} + \frac{5}{2}\delta_T^2. \tag{10}$$

Choosing $\eta = \sqrt{\Delta_f b/(L\sigma^2 T)}$ gives a clean rate, $\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\|\nabla f(\mathbf{w}_t)\|^2 \le 4\sqrt{\frac{\Delta_f L\sigma^2}{bT}} + \frac{5}{2}\delta_T^2$.

Run training long enough and the first term disappears on its own, at the usual $\mathcal{O}(1/\sqrt{bT})$ pace. What is left is $\frac{5}{2}\delta_T^2$, and it does not go away with more steps. It only shrinks if GhostMatch's selected subsets get better at representing the batches they come from.

Discussion. Setting $\mathsf{S}_t = \mathsf{B}_t$ sends δ_T^2 to zero and recovers exactly the ordinary nonconvex mini-batch SGD bound (Ghadimi & Lan, 2013; Bottou et al., 2018), confirming GhostMatch costs nothing beyond what its own representation error predicts. This is the point of the section: the same $c_t(\mathsf{S}_t)$ that Section 5.2 keeps small combinatorially is the exact quantity that determines GhostMatch's convergence floor here. Unlike CoLM's variance-reduction analysis, which shows the coreset's gradient has lower variance than a random subset of the same size, this bound substitutes that combinatorial quantity directly into the general nonconvex stationarity bound. The bound above targets plain SGD; our experiments also run GhostMatch with adaptive optimizers, supplying $\widehat{\mathbf{g}}_t$ wherever the optimizer would otherwise use the full-batch gradient.

A stronger guarantee under additional structure. The rate above is sublinear, as is standard for nonconvex objectives. If f additionally satisfies the Polyak–Łojasiewicz condition, Appendix C shows GhostMatch converges *linearly* instead, to a neighborhood whose size is set by the same two error sources, sampling variance and representation cost, that appear throughout this section.

7 EXPERIMENTS

Our experiments address four questions raised by GhostMatch's design: can a selector using only the current candidate batch compete with methods that additionally observe validation examples? Does exact gradient geometry matter for selection quality? Can that geometry be computed efficiently at LLM scale? And when can selecting a subset be disadvantageous at a fixed training horizon? Section 5 shows that GhostMatch approximately minimizes a representation cost, while Section 6 connects that cost to optimization error; the experiments test whether these guarantees translate into useful selection in practice.

7.1 EXPERIMENTAL SETUP

We fine-tune TinyLlama-1.1B (Zhang et al., 2024), Llama-2-7B (Touvron et al., 2023), and Mistral-7B (Jiang et al., 2023) on LESS (Xia et al., 2024), and Llama-3-8B (AI@Meta, 2024) on Alpaca (Taori et al., 2023), evaluating on MMLU (Hendrycks et al., 2020), TyDiQA (Clark et al., 2020), and SamSum (Gliwa et al., 2019). Full model–training–evaluation configurations are given in Appendix E.1, with additional experimental results in Appendix E and extended-horizon robustness checks in Appendix F.

Table 1: Main results across four configurations, mean ± std over seeds (TinyLlama-1.1B and Llama-2-7B: 2 subjects × 3 seeds, n=6; Mistral-7B and Llama-3-8B: n=2). Lower is better for perplexity; higher is better for F1 and ROUGE-1. Bold marks the best method per column.

	TinyLlama-1.1B MMLU ppl	Llama-2-7B MMLU ppl	Mistral-7B TyDiQA F1	Llama-3-8B SamSum R1
Regular	6.48±0.82	9.51±0.59	0.507±0.000	**0.174±0.004**
GREATS	**4.98±0.20**	9.63±0.46	**0.648±0.005**	0.132±0.001
GhostMatch	5.99±0.62	9.73±0.55	0.609±0.026	0.090±0.006
OPUS	5.12±0.15	**9.38±0.56**	0.572±0.004	0.156±0.008
ZeroOrder	6.15±0.74	9.95±0.46	0.585±0.007	0.036±0.009

Our baselines disentangle the effects of subset selection, access to a held-out signal, and gradient approximation. **Regular** trains on the full candidate batch. **GREATS** (Wang et al., 2024) approximates the one-step reduction in validation loss by Taylor expansion and greedily selects examples using validation-gradient alignment with a redundancy correction; we compute the required inner products with the same ghost machinery used by GhostMatch. Appendix D shows that its approximate set objective need not be submodular. **OPUS** (Wang et al., 2026), originally developed for pretraining, uses a proxy-directed, redundancy-aware selection objective. We adapt it to our finetuning setting using the same five held-out proxy examples as GREATS, exact candidate-gradient interactions, an identity preconditioner for selection, and greedy top-k selection. **ZeroOrder** is a controlled, validation-free ablation inspired by CoLM's zeroth-order gradient estimation (Nguyen et al., 2025): it replaces GhostMatch's exact ghost-computed gradient geometry with a zeroth-order estimate while keeping GhostMatch's objective and selection rule unchanged. For a controlled comparison, GREATS and OPUS use the same 5 held-out examples, whereas GhostMatch and ZeroOrder use none. All selection methods choose k=4 examples from a candidate pool of $2k$; Regular trains on all $2k$.

7.2 Validation-Free Selection

Table 1 reports downstream performance. Despite having no validation access, GhostMatch remains competitive on three of four configurations while training on half the candidate data. On TinyLlama-1.1B it reaches 5.99 perplexity, better than Regular (6.48) and ZeroOrder (6.15), though behind GREATS (4.98) and OPUS (5.12). Llama-2-7B is tightly grouped, with perplexities ranging from 9.38 to 9.95. On Mistral-7B, GhostMatch reaches 0.609 F1, above Regular (0.507), OPUS (0.572), and ZeroOrder (0.585), and within 0.040 of GREATS (0.648). Thus, in this configuration, representing the candidate batch directly in gradient space outperforms a validation-directed selector despite using no held-out examples. Appendix E.3 reports complementary task metrics, including standard MMLU accuracy. On Llama-3-8B, Regular performs best and every selection method underperforms it, including GREATS and OPUS, indicating a regime in which subset selection is premature at the 300-step training horizon. Section 7.5 examines this case, while Appendix F shows that this early disadvantage does not persist under longer training.

7.3 Why Does Exact Gradient Geometry Matter?

Corollary 1 shows that gradient-representation error enters directly into the stationarity bound, so inaccurate geometry can lead to subsets that only appear representative. ZeroOrder isolates this effect while holding the selection objective and rule fixed. GhostMatch outperforms it on every configuration: 5.99 vs. 6.15 perplexity (TinyLlama-1.1B), 9.73 vs. 9.95 (Llama-2-7B), 0.609 vs. 0.585 F1 (Mistral-7B), and 0.090 vs. 0.036 ROUGE-1 (Llama-3-8B). Appendix E.2 shows that this advantage holds in all $16/16$ matched final evaluations, while Appendix E.4 shows that it persists in $39/40$ available checkpoint-level comparisons. These results are consistent with the convergence analysis, showing that more faithful gradient geometry produces better selected updates and, in these experiments, better downstream performance.

7.4 Cost of Exact Gradient Geometry

Figure 1 measures the computational cost of this exactness. The scoring microbenchmark uses Phi-2 (Microsoft Research, 2023), architecture-matched to CoLM's zeroth-order scorer (Nguyen et al., 2025), avoiding a model-size confound.

For the exact-gradient comparison, we report the shared feasible range $b \in \{8, 16, 24\}$ on a single L40S, over which GhostMatch is consistently faster than naive per-example gradient materialization. The CoLM-style zeroth-order scorer is also shown through $b{=}64$ as a reference for the computational advantage of approximate geometry. Thus, ghost inner products substantially reduce the cost of preserving exact gradient geometry, while the controlled ZeroOrder comparison in Section 7.3 quantifies the downstream trade-off incurred by replacing it with a zeroth-order approximation.

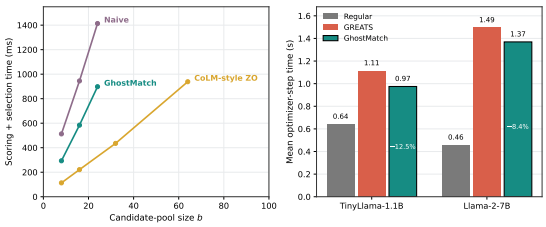


Figure 1: Left (*scoring microbenchmark*): scoring-plus-selection time vs. candidate-pool size b on Phi-2, comparing naive full-gradient scoring, GhostMatch's ghost inner products, and a CoLM-style zeroth-order approximation (Nguyen et al., 2025). Right (*end-to-end training*): mean optimizer-step wall-clock time for Regular, GREATS, and GhostMatch on TinyLlama-1.1B and Llama-2-7B. The percentage inside each GhostMatch bar is its reduction in step time relative to GREATS at the same model and candidate-pool size (e.g., -12.5% means 12.5% lower step time).

End-to-end timing shows a similar computational advantage over validation-directed selection. GhostMatch increases mean step time relative to Regular from 0.64 to 0.97 s on TinyLlama-1.1B and from 0.46 to 1.37 s on Llama-2-7B, but remains 12.5% and 8.4% faster than GREATS, respectively, at the same candidate-pool size. Both methods use ghost inner products, while GREATS additionally requires computing a validation-gradient signal at each selection step.

7.5 A Boundary Case: When Selection Initially Hurts

On Llama-3-8B, Regular reaches 0.174 ROUGE-1, ahead of OPUS (0.156), GREATS (0.132), GhostMatch (0.090), and ZeroOrder (0.036), so every method training on half the candidate pool underperforms full-batch training at the 300-step horizon. The model is fine-tuned from a raw checkpoint using a chat template it has not previously seen, suggesting a cold-start regime in which learning the output format remains important. In this regime, more data can outweigh better-chosen data, and Regular's full $2k$ candidates outperform every k-sized subset. That GREATS and OPUS also underperform Regular despite validation access supports this interpretation and identifies a regime in which aggressive subset selection is not yet beneficial at this training budget. Extended-horizon results in Appendix F show that Regular's early advantage disappears under longer training, indicating that this boundary case is transient rather than a persistent limitation of subset selection.

8 Conclusion and Limitations

We introduced GhostMatch, a validation free method for online batch selection that replaces a candidate batch with a weighted subset whose gradients preserve the batch's geometry. Casting this selection problem as facility location yields a monotone submodular objective and gives greedy selection the classical $(1 - 1/e)$ approximation guarantee. We then derived a nonconvex stationarity bound in which the resulting gradient representation error appears explicitly, establishing a direct link between the combinatorial objective optimized by GhostMatch and the optimization error induced by subset training. Finally, ghost inner products provide the required gradient geometry without materializing model sized gradients for individual examples, making exact gradient based selection practical at LLM scale.

The current analysis has several important boundaries. Our convergence results characterize the optimization effect of gradient representation error under SGD-style updates and do not provide a direct guarantee on downstream generalization or on the full dynamics of adaptive optimizers such as AdamW, which is used in our experiments. GhostMatch also optimizes representativeness within the current candidate batch, so its benefit depends on that batch containing useful information to preserve. Finally, although ghost computation avoids materializing model sized gradients for individual examples, constructing exact pairwise gradient geometry still introduces memory and computation that grow with the candidate pool. These limitations motivate extensions to adaptive optimizers, larger candidate pools, and selection objectives that incorporate broader training dynamics.

ETHICS STATEMENT

This work studies algorithmic methods for online batch selection in LLM fine-tuning and does not involve human subjects, the collection of new personal data, or deployment in high-stakes decision-making settings. The experiments use existing datasets and pretrained language models described in Section 7. As with other methods that improve the efficiency of LLM training, GhostMatch may be used within systems that inherit biases, inaccuracies, or other risks from their underlying models and training data. Our work focuses on the batch-selection mechanism itself and does not evaluate downstream deployment in safety-critical settings.

AI USE STATEMENT

AI systems assisted with drafting and polishing the exposition, carrying out computational and logical checks, suggesting ideas, and locating references. The authors independently verified the technical claims, proofs, experimental results, and citations and take full responsibility for the contents of the paper.

REPRODUCIBILITY STATEMENT

An anonymized implementation of GhostMatch, together with the experiment scripts, result summaries, timing data, and code used to reproduce the reported figures, is available at `https://anonymous.4open.science/r/ghostmatch-review-E1C7/`. Section 3 defines the optimization setting and gradient-matching formulation. Section 4 describes the computation of pairwise gradient geometry, while Section 5 gives the representative weighting rule, facility-location objective, and greedy selection procedure. Appendix A provides the complete GhostMatch algorithm. The assumptions and convergence results are stated in Section 6, with complete proofs in Appendix B and additional results in Appendix C. Section 7 describes the experimental setup, baselines, and evaluation protocol, while Appendix E provides the detailed experimental configurations, ablations, and additional metrics.

REFERENCES

AI@Meta. Llama 3 model card, 2024.

Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *SIAM review*, 60(2):223–311, 2018.

Jonathan H Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. Tydi qa: A benchmark for information-seeking question answering in ty pologically di verse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470, 2020.

Gerard Cornuejols, Marshall Fisher, and George L Nemhauser. On the uncapacitated location problem. In *Annals of Discrete Mathematics*, volume 1, pp. 163–177. Elsevier, 1977.

Saeed Ghadimi and Guanghui Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM journal on optimization*, 23(4):2341–2368, 2013.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. Samsum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pp. 70–79, 2019.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL `https://arxiv.org/abs/2310.06825`.

Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In *Joint European conference on machine learning and knowledge discovery in databases*, pp. 795–811. Springer, 2016.

Angelos Katharopoulos and François Fleuret. Not all samples are created equal: Deep learning with importance sampling. In *International conference on machine learning*, pp. 2525–2534. PMLR, 2018.

Krishnateja Killamsetty, Sivasubramanian Durga, Ganesh Ramakrishnan, Abir De, and Rishabh Iyer. Grad-match: Gradient matching based data subset selection for efficient deep model training. In *International Conference on Machine Learning*, pp. 5464–5474. PMLR, 2021.

Ilya Loshchilov and Frank Hutter. Online batch selection for faster training of neural networks. *arXiv preprint arXiv:1511.06343*, 2015.

Microsoft Research. Phi-2: The surprising power of small language models. `https://www.microsoft.com/en-us/research/blog/phi-2-the-surprising-power-of-small-language-models/`, 2023.

Baharan Mirzasoleiman, Ashwinkumar Badanidiyuru, Amin Karbasi, Jan Vondrák, and Andreas Krause. Lazier than lazy greedy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2015.

Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. Coresets for data-efficient training of machine learning models. In *International Conference on Machine Learning*, pp. 6950–6960. PMLR, 2020.

George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions – I. *Mathematical programming*, 14(1):265–294, 1978.

Dang Nguyen, Wenhan Yang, Rathul Anand, Yu Yang, and Baharan Mirzasoleiman. Mini-batch coresets for memory-efficient language model training on data mixtures. In *International Conference on Learning Representations*, volume 2025, pp. 82476–82496, 2025.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Stanford alpaca: An instruction-following llama model, 2023.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Jiachen T Wang, Tong Wu, Dawn Song, Prateek Mittal, and Ruoxi Jia. Greats: Online selection of high-quality data for llm training in every iteration. *Advances in Neural Information Processing Systems*, 37:131197–131223, 2024.

Shaobo Wang, Xuan Ouyang, Tianyi Xu, Yuzheng Hu, Jialin Liu, Guo Chen, Tianyu Zhang, Junhao Zheng, Kexin Yang, Xingzhang Ren, et al. Opus: towards efficient and principled data selection in large language model pre-training in every iteration. *arXiv preprint arXiv:2602.05400*, 2026.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. Less: Selecting influential data for targeted instruction tuning. *arXiv preprint arXiv:2402.04333*, 2024.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. Tinyllama: An open-source small language model. *arXiv preprint arXiv:2401.02385*, 2024.

APPENDIX OVERVIEW

The appendix is organized as follows.

- Appendix A presents the complete GhostMatch algorithm and its stochastic-greedy variant for larger candidate batches.
- Appendix B proves the one-step descent bound, the general nonconvex convergence theorem, and its specialization to GhostMatch.
- Appendix C gives additional theoretical results.
  - Appendix C.1 establishes convergence under the Polyak–Łojasiewicz condition.
  - Appendix C.2 establishes asymptotic stationarity with diminishing step sizes.
  - Appendix C.3 connects the facility-location approximation guarantee to the representation error appearing in the convergence analysis.
- Appendix D gives a counterexample showing that the GREATS Taylor surrogate is not submodular in general.
- Appendix E contains additional experimental results.
  - Appendix E.1 summarizes the four experimental configurations.
  - Appendix E.2 reports the full exactness ablation.
  - Appendix E.3 reports complementary evaluation metrics.
  - Appendix E.4 studies the exact-gradient advantage throughout training.
  - Appendix F reports extended-horizon robustness checks.

A THE GHOSTMATCH ALGORITHM

Algorithm 1 makes the selection and update procedure described in Section 5.3 fully explicit, step by step, from the ghost-computed Gram matrix through to the final parameter update.

Algorithm 1 GHOSTMATCH: Validation-Free Gradient Coreset Selection

Input: current parameters $\mathbf{w}_t$, candidate batch $\mathsf{B}_t = \{z_1, \dots, z_b\}$, subset size k
Compute the candidate-gradient Gram matrix $\mathbf{K}_t$ using ghost inner products
Compute $d_{t,ij} = \sqrt{(\mathbf{K}_t)_{ii} + (\mathbf{K}_t)_{jj} - 2(\mathbf{K}_t)_{ij}}$ for all $i, j \in [b]$
Set $m_t = \max_{i,j\in[b]} d_{t,ij}$ and $s_{t,ij} = m_t - d_{t,ij}$ ▷ convert distances to similarities
Define $F_t(\mathsf{S}) := \sum_{i=1}^{b} \max_{j:\, z_j \in \mathsf{S}} s_{t,ij}$ for any $\mathsf{S} \subseteq \mathsf{B}_t$, with $F_t(\emptyset) := 0$ ▷ see equation 4
Initialize $\mathsf{S} \leftarrow \emptyset$
for $r = 1, \dots, k$ **do** ▷ greedy facility-location selection
    Select $j^\star \in \arg\max_{j:\, z_j \in \mathsf{B}_t \setminus \mathsf{S}} \big[F_t(\mathsf{S} \cup \{z_j\}) - F_t(\mathsf{S})\big]$
    $\mathsf{S} \leftarrow \mathsf{S} \cup \{z_{j^\star}\}$
end for
for $i = 1, \dots, b$ **do** ▷ assign each candidate to its nearest selected example
    $\sigma_{\mathsf{S}}(i) \in \arg\min_{j:\, z_j \in \mathsf{S}} d_{t,ij}$
end for
for each $z_j \in \mathsf{S}$ **do** ▷ weight = number of candidates represented
    $\gamma_{t,j} \leftarrow |\{i \in [b] : \sigma_{\mathsf{S}}(i) = j\}|$
end for
Compute $\widehat{\ell}_t = \frac{1}{b} \sum_{z_j \in \mathsf{S}_t} \gamma_{t,j} \ell(\mathbf{w}_t, z_j)$ and obtain $\widehat{\mathbf{g}}_t = \nabla_{\mathbf{w}} \widehat{\ell}_t$ via backpropagation on $\widehat{\ell}_t$
Compute $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \widehat{\mathbf{g}}_t$ for SGD (or the analogous update for the chosen optimizer)
Output: updated parameters $\mathbf{w}_{t+1}$, selected subset $\mathsf{S}_t = \mathsf{S}$, and weights $\{\gamma_{t,j}\}_{z_j \in \mathsf{S}_t}$

A.1 STOCHASTIC-GREEDY VARIANT FOR LARGER CANDIDATE BATCHES

For larger candidate batches, the exact greedy search in Algorithm 1 can be replaced by stochastic greedy selection (Mirzasoleiman et al., 2015). Rather than evaluating the marginal gain of every remaining candidate at each greedy step, stochastic greedy evaluates only a randomly sampled subset. We apply this procedure directly to GhostMatch's facility-location objective F_t; all subsequent representative assignment, weighting, and parameter-update steps remain unchanged.

Algorithm 2 GHOSTMATCH with Stochastic-Greedy Selection

Input: current parameters $\mathbf{w}_t$, candidate batch $\mathsf{B}_t = \{z_1, \dots, z_b\}$, subset size k, accuracy parameter $\varepsilon \in (0,1)$
Compute the candidate-gradient Gram matrix $\mathbf{K}_t$ using ghost inner products
Compute $d_{t,ij} = \sqrt{(\mathbf{K}_t)_{ii} + (\mathbf{K}_t)_{jj} - 2(\mathbf{K}_t)_{ij}}$ for all $i, j \in [b]$
Set $m_t = \max_{i,j\in[b]} d_{t,ij}$ and $s_{t,ij} = m_t - d_{t,ij}$
Define $F_t(\mathsf{S}) := \sum_{i=1}^{b} \max_{j:\, z_j \in \mathsf{S}} s_{t,ij}$ for nonempty $\mathsf{S} \subseteq \mathsf{B}_t$, with $F_t(\emptyset) := 0$
Set $q \leftarrow \lceil \frac{b}{k} \log \frac{1}{\varepsilon} \rceil$
Initialize $\mathsf{S} \leftarrow \emptyset$
for $r = 1, \dots, k$ **do**
    Sample uniformly at random $\mathsf{R}_r \subseteq \mathsf{B}_t \setminus \mathsf{S}$ with $|\mathsf{R}_r| = \min\{q, |\mathsf{B}_t \setminus \mathsf{S}|\}$
    Select $z_{j^\star} \in \arg\max_{z_j \in \mathsf{R}_r} [F_t(\mathsf{S} \cup \{z_j\}) - F_t(\mathsf{S})]$
    $\mathsf{S} \leftarrow \mathsf{S} \cup \{z_{j^\star}\}$
end for
for $i = 1, \dots, b$ **do**
    $\sigma_{\mathsf{S}}(i) \in \arg\min_{j:\, z_j \in \mathsf{S}} d_{t,ij}$
end for
for each $z_j \in \mathsf{S}$ **do**
    $\gamma_{t,j} \leftarrow |\{i \in [b] : \sigma_{\mathsf{S}}(i) = j\}|$
end for
Compute $\widehat{\ell}_t = \frac{1}{b} \sum_{z_j \in \mathsf{S}} \gamma_{t,j} \ell(\mathbf{w}_t, z_j)$ and obtain $\widehat{\mathbf{g}}_t = \nabla_{\mathbf{w}} \widehat{\ell}_t$ by backpropagation
Compute $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \widehat{\mathbf{g}}_t$ for SGD, or the analogous update for the chosen optimizer
Output: updated parameters $\mathbf{w}_{t+1}$, selected subset $\mathsf{S}_t = \mathsf{S}$, and weights $\{\gamma_{t,j}\}_{z_j \in \mathsf{S}_t}$

Let $\varepsilon \in (0,1)$ control the approximation-computation tradeoff and set $q = \lceil \frac{b}{k} \log \frac{1}{\varepsilon} \rceil$.

At each selection step, we sample at most q currently unselected candidates and evaluate the facility-location marginal gain only over this sampled set.

Because F_t is nonnegative, monotone, and submodular, the standard stochastic-greedy result of Mirzasoleiman et al. (2015) applies directly. If S^{SG} denotes the subset returned by Algorithm 2 and $\mathsf{S}^\star$ is an optimal size-k subset, then

$$\mathbb{E}\big[F_t(\mathsf{S}^{\mathrm{SG}})\big] \geq \left(1 - \frac{1}{e} - \varepsilon\right) F_t(\mathsf{S}^\star).$$

The stochastic-greedy stage requires $O(b \log(1/\varepsilon))$ marginal-gain evaluations, rather than evaluating every remaining candidate at every greedy iteration (Mirzasoleiman et al., 2015). Thus, it provides a drop-in alternative to the exact greedy stage of GhostMatch when the candidate batch becomes large, while the gradient representation and weighted training update remain unchanged.

B PROOFS FOR SECTION 6

This appendix proves, in order, the three results stated in Section 6: the one-step descent bound (Lemma 1), its accumulation into a stationarity guarantee (Theorem 1), and its specialization to GhostMatch (Corollary 1). Each proof builds directly on the one before it, so we present them in the same order the results appear in the main text.

Proof of Lemma 1. The starting point is L-smoothness applied to a single step of equation 6, which bounds how much the loss can change in terms of the update direction actually taken,

$$f(\mathbf{w}_{t+1}) \leq f(\mathbf{w}_t) - \eta_t \langle \nabla f(\mathbf{w}_t), \widehat{\mathbf{g}}_t \rangle + \frac{L\eta_t^2}{2} \|\widehat{\mathbf{g}}_t\|^2.$$

Writing $\widehat{\mathbf{g}}_t = \overline{\mathbf{g}}_t + \mathbf{e}_t$ and taking conditional expectation lets us isolate the true gradient term using Assumption 2,

$$\mathbb{E}_t[f(\mathbf{w}_{t+1})] \leq\ f(\mathbf{w}_t) - \eta_t \|\nabla f(\mathbf{w}_t)\|^2 - \eta_t \mathbb{E}_t \langle \nabla f(\mathbf{w}_t), \mathbf{e}_t \rangle + \frac{L\eta_t^2}{2} \mathbb{E}_t \|\overline{\mathbf{g}}_t + \mathbf{e}_t\|^2.$$

Two standard inequalities clean up the cross term and the norm term. Cauchy–Schwarz together with Young's inequality gives

$$-\langle \nabla f(\mathbf{w}_t), \mathbf{e}_t \rangle \le \frac{1}{4}\|\nabla f(\mathbf{w}_t)\|^2 + \|\mathbf{e}_t\|^2,$$

and expanding the squared norm gives $\|\overline{\mathbf{g}}_t + \mathbf{e}_t\|^2 \le 2\|\overline{\mathbf{g}}_t\|^2 + 2\|\mathbf{e}_t\|^2$. Conditional unbiasedness (Assumption 2) turns $\mathbb{E}_t\|\overline{\mathbf{g}}_t\|^2$ into a term we can bound directly,

$$\mathbb{E}_t\|\overline{\mathbf{g}}_t\|^2 = \|\nabla f(\mathbf{w}_t)\|^2 + \mathbb{E}_t\|\overline{\mathbf{g}}_t - \nabla f(\mathbf{w}_t)\|^2 \le \|\nabla f(\mathbf{w}_t)\|^2 + \frac{\sigma^2}{b}.$$

Substituting all three bounds back in collects every $\|\nabla f(\mathbf{w}_t)\|^2$ and every $\mathbb{E}_t[\|\mathbf{e}_t\|^2]$ term on one side,

$$\mathbb{E}_t[f(\mathbf{w}_{t+1})] \le\ f(\mathbf{w}_t) - \left(\frac{3\eta_t}{4} - L\eta_t^2\right)\|\nabla f(\mathbf{w}_t)\|^2 + \frac{L\eta_t^2\sigma^2}{b} + (\eta_t + L\eta_t^2)\mathbb{E}_t[\|\mathbf{e}_t\|^2].$$

The step-size restriction $\eta_t \le 1/(4L)$ is exactly what is needed to simplify the two remaining coefficients: it gives $3\eta_t/4 - L\eta_t^2 \ge \eta_t/2$ and $\eta_t + L\eta_t^2 \le 5\eta_t/4$, and substituting these two facts in proves equation 7. □

Proof of Theorem 1. The one-step bound above holds at every iteration. Summing it over $t = 0, \dots, T-1$, after taking total expectation via the tower property, gives

$$\frac{1}{2}\sum_{t=0}^{T-1}\eta_t\,\mathbb{E}\|\nabla f(\mathbf{w}_t)\|^2 \le\ f(\mathbf{w}_0) - \mathbb{E}[f(\mathbf{w}_T)] + \frac{L\sigma^2}{b}\sum_{t=0}^{T-1}\eta_t^2 + \frac{5}{4}\sum_{t=0}^{T-1}\eta_t\,\mathbb{E}\|\mathbf{e}_t\|^2.$$

This uses the fact that the loss terms on the right of the one-step bound telescope: summing $\mathbb{E}[f(\mathbf{w}_t)] - \mathbb{E}[f(\mathbf{w}_{t+1})]$ over consecutive iterations cancels every intermediate term, leaving only $f(\mathbf{w}_0) - \mathbb{E}[f(\mathbf{w}_T)]$. Since $f(\mathbf{w}_T) \ge f_\star$ by Assumption 1, this quantity is bounded by $\Delta_f := f(\mathbf{w}_0) - f_\star$. Dividing through by $\frac{1}{2}\sum_{t=0}^{T-1}\eta_t$ gives equation 8. □

Proof of Corollary 1. This is where GhostMatch's own guarantee enters. Everything up to Theorem 1 holds for an arbitrary selection error $\mathbf{e}_t$; GhostMatch's contribution is supplying an explicit bound on that error through equation 5, $\|\mathbf{e}_t\|^2 \le c_t(\mathsf{S}_t)^2/b^2$, which holds for every realization of the candidate batch, not just in expectation. Substituting this bound directly into Theorem 1 in place of $\mathbb{E}\|\mathbf{e}_t\|^2$ yields equation 9. □

C Additional Convergence Results

This appendix develops several consequences of the convergence analysis in Section 6. We first strengthen the convergence result under the Polyak–Łojasiewicz condition. We then establish asymptotic stationarity under diminishing step sizes and finally connect the facility-location approximation guarantee to the representation cost that appears in the GhostMatch convergence bound.

C.1 Convergence Under the Polyak–Łojasiewicz Condition

We first consider objectives satisfying the Polyak–Łojasiewicz condition Karimi et al. (2016).

Assumption 3 (Polyak–Łojasiewicz condition) *There exists $\mu > 0$ such that, for every $\mathbf{w}$,*

$$\|\nabla f(\mathbf{w})\|^2 \ge 2\mu\big(f(\mathbf{w}) - f_\star\big).$$

The PL condition does not require f to be convex. It instead relates the gradient norm directly to objective suboptimality. Combined with Lemma 1, this converts the one-step descent bound into a contraction up to the two error sources already identified in Section 6, candidate-batch sampling and subset representation error.

Recall from equation 5 that

$$\|\mathbf{e}_t\| \le \rho_t, \qquad \rho_t := \frac{c_t(\mathsf{S}_t)}{b}.$$

Theorem 2 (PL convergence of GhostMatch) *Suppose Assumptions 1, 2, and 3 hold. Assume that GhostMatch uses a constant step size satisfying*

$$0 < \eta \leq \min\left\{\frac{1}{4L}, \frac{1}{\mu}\right\},$$

and that its normalized representation error satisfies

$$\mathbb{E}\big[\rho_t^2\big] \leq \delta^2 \qquad \textit{for all } t.$$

Then

$$\mathbb{E}\big[f(\mathbf{w}_T) - f_\star\big] \leq \ (1-\mu\eta)^T \Delta_f + \big(1-(1-\mu\eta)^T\big)\left(\frac{L\eta\sigma^2}{\mu b} + \frac{5\delta^2}{4\mu}\right), \tag{11}$$

where $\Delta_f = f(\mathbf{w}_0) - f_\star$. Consequently,

$$\limsup_{T\to\infty} \mathbb{E}\big[f(\mathbf{w}_T) - f_\star\big] \leq \frac{L\eta\sigma^2}{\mu b} + \frac{5\delta^2}{4\mu}. \tag{12}$$

Proof. Lemma 1 gives

$$\mathbb{E}_t[f(\mathbf{w}_{t+1})] \leq f(\mathbf{w}_t) - \frac{\eta}{2}\|\nabla f(\mathbf{w}_t)\|^2 + \frac{L\eta^2\sigma^2}{b} + \frac{5\eta}{4}\mathbb{E}_t\big[\|\mathbf{e}_t\|^2\big].$$

Since $\|\mathbf{e}_t\| \leq \rho_t$,

$$\mathbb{E}_t[f(\mathbf{w}_{t+1})] \leq f(\mathbf{w}_t) - \frac{\eta}{2}\|\nabla f(\mathbf{w}_t)\|^2 + \frac{L\eta^2\sigma^2}{b} + \frac{5\eta}{4}\mathbb{E}_t[\rho_t^2].$$

Assumption 3 implies

$$\frac{1}{2}\|\nabla f(\mathbf{w}_t)\|^2 \geq \mu\big(f(\mathbf{w}_t) - f_\star\big),$$

and therefore

$$-\frac{\eta}{2}\|\nabla f(\mathbf{w}_t)\|^2 \leq -\mu\eta\big(f(\mathbf{w}_t) - f_\star\big).$$

Substituting this inequality into the one-step bound and subtracting $f_\star$ from both sides gives

$$\mathbb{E}_t\big[f(\mathbf{w}_{t+1}) - f_\star\big] \leq \ (1-\mu\eta)\big(f(\mathbf{w}_t) - f_\star\big) + \frac{L\eta^2\sigma^2}{b} + \frac{5\eta}{4}\mathbb{E}_t[\rho_t^2].$$

Taking total expectation and using $\mathbb{E}[\rho_t^2] \leq \delta^2$ yields

$$\mathbb{E}\big[f(\mathbf{w}_{t+1}) - f_\star\big] \leq \ (1-\mu\eta)\mathbb{E}\big[f(\mathbf{w}_t) - f_\star\big] + \frac{L\eta^2\sigma^2}{b} + \frac{5\eta}{4}\delta^2.$$

Define

$$q := 1-\mu\eta, \qquad r := \frac{L\eta^2\sigma^2}{b} + \frac{5\eta}{4}\delta^2.$$

The step-size assumption ensures $0 \leq q < 1$. Repeated application of the preceding recursion therefore gives

$$\mathbb{E}\big[f(\mathbf{w}_T) - f_\star\big] \leq q^T\Delta_f + r\sum_{r'=0}^{T-1} q^{r'}.$$

The geometric sum satisfies

$$\sum_{r'=0}^{T-1} q^{r'} = \frac{1-q^T}{1-q} = \frac{1-(1-\mu\eta)^T}{\mu\eta}.$$

Substituting this identity and the definition of r gives equation 11. Since $0 \leq 1-\mu\eta < 1$, the geometric factor $(1-\mu\eta)^T$ converges to zero as $T \to \infty$, which gives equation 12. □

Theorem 2 shows that GhostMatch converges geometrically to a neighborhood of the optimum under the PL condition. Candidate-batch stochasticity contributes the term $L\eta\sigma^2/(\mu b)$, while persistent representation error contributes $5\delta^2/(4\mu)$. If the selection error vanishes, the second term disappears and the result reduces to the standard constant-step stochastic gradient behavior under the PL condition (Karimi et al., 2016).

C.2 Asymptotic Stationarity with Diminishing Step Sizes

The constant-step bound in equation 10 contains a persistent contribution from the representation error. We next show that this contribution vanishes asymptotically when the step sizes decrease in the standard stochastic approximation regime and the weighted representation error also vanishes.

Corollary 2 (Asymptotic stationarity) *Suppose Assumptions 1 and 2 hold. Let $\{\mathbf{w}_t\}_{t\geq 0}$ be generated by GhostMatch with deterministic step sizes satisfying*

$$0 < \eta_t \leq \frac{1}{4L}, \qquad \sum_{t=0}^{\infty} \eta_t = \infty, \qquad \sum_{t=0}^{\infty} \eta_t^2 < \infty.$$

If

$$\frac{\sum_{t=0}^{T-1} \eta_t \, \mathbb{E}[\rho_t^2]}{\sum_{t=0}^{T-1} \eta_t} \longrightarrow 0 \qquad \textit{as} \quad T \to \infty, \tag{13}$$

then

$$\frac{\sum_{t=0}^{T-1} \eta_t \, \mathbb{E}\|\nabla f(\mathbf{w}_t)\|^2}{\sum_{t=0}^{T-1} \eta_t} \longrightarrow 0. \tag{14}$$

Proof. Corollary 1 and $\rho_t = c_t(\mathsf{S}_t)/b$ give

$$\frac{\sum_{t=0}^{T-1} \eta_t \, \mathbb{E}\|\nabla f(\mathbf{w}_t)\|^2}{\sum_{t=0}^{T-1} \eta_t} \leq \frac{2\Delta_f}{\sum_{t=0}^{T-1} \eta_t} + \frac{2L\sigma^2}{b} \frac{\sum_{t=0}^{T-1} \eta_t^2}{\sum_{t=0}^{T-1} \eta_t} + \frac{5}{2} \frac{\sum_{t=0}^{T-1} \eta_t \, \mathbb{E}[\rho_t^2]}{\sum_{t=0}^{T-1} \eta_t}.$$

Each term on the right converges to zero. The first term vanishes because $\sum_t \eta_t = \infty$. For the second term, $\sum_t \eta_t^2 < \infty$ implies that its numerator converges to a finite value, while $\sum_t \eta_t = \infty$ makes its denominator diverge. The third term vanishes by assumption equation 13. The entire right-hand side therefore converges to zero. The left-hand side is nonnegative, so the squeeze theorem gives equation 14. □

Condition equation 13 holds, in particular, whenever $\mathbb{E}[\rho_t^2] \longrightarrow 0$.

Indeed, a sequence converging to zero also has a weighted average converging to zero when the weights are positive and their cumulative sum diverges.

The stationarity statement also has the standard random-iterate interpretation. Let $R_T \in \{0, \ldots, T-1\}$ be drawn according to

$$\mathbb{P}(R_T = t) = \frac{\eta_t}{\sum_{s=0}^{T-1} \eta_s}.$$

Then

$$\mathbb{E}\|\nabla f(\mathbf{w}_{R_T})\|^2 = \sum_{t=0}^{T-1} \frac{\eta_t}{\sum_{s=0}^{T-1} \eta_s} \mathbb{E}\|\nabla f(\mathbf{w}_t)\|^2 = \frac{\sum_{t=0}^{T-1} \eta_t \, \mathbb{E}\|\nabla f(\mathbf{w}_t)\|^2}{\sum_{t=0}^{T-1} \eta_t} \longrightarrow 0.$$

Thus, when the gradient representation error becomes asymptotically negligible, GhostMatch recovers first-order stationarity in the usual weighted-iterate sense.

C.3 From the Submodular Guarantee to Representation Error

The convergence results depend on the representation cost $c_t(\mathsf{S}_t)$ achieved by the selected subset. We now relate this cost to the classical approximation guarantee for greedy facility-location maximization.

Let

$$\mathsf{S}_t^\star \in \arg\min_{\substack{\mathsf{S} \subseteq \mathsf{B}_t \\ |\mathsf{S}| = k}} c_t(\mathsf{S}),$$

denote an optimal size-k subset for the representation cost, where $k \geq 1$, and let $\mathsf{S}_t^{\mathrm{G}}$ denote the size-k subset returned by greedy maximization of F_t. Because $F_t(\mathsf{S}) = bm_t - c_t(\mathsf{S})$, for every nonempty subset, $\mathsf{S}_t^\star$ also maximizes F_t among all size-k subsets.

Proposition 2 (Representation cost of greedy selection) *The greedy subset satisfies*

$$c_t(\mathsf{S}_t^{\mathrm{G}}) \leq \frac{bm_t}{e} + \left(1 - \frac{1}{e}\right) c_t(\mathsf{S}_t^\star). \tag{15}$$

Proof. Since F_t is normalized, monotone, and submodular, the classical greedy guarantee under a cardinality constraint gives Nemhauser et al. (1978)

$$F_t(\mathsf{S}_t^{\mathrm{G}}) \geq \left(1 - \frac{1}{e}\right) F_t(\mathsf{S}_t^{\star}).$$

Using $F_t(\mathsf{S}) = bm_t - c_t(\mathsf{S})$ for the two nonempty size-k subsets gives

$$bm_t - c_t(\mathsf{S}_t^{\mathrm{G}}) \geq \left(1 - \frac{1}{e}\right) (bm_t - c_t(\mathsf{S}_t^{\star})).$$

Rearranging,

$$c_t(\mathsf{S}_t^{\mathrm{G}}) \leq bm_t - \left(1 - \frac{1}{e}\right) (bm_t - c_t(\mathsf{S}_t^{\star})) = \frac{bm_t}{e} + \left(1 - \frac{1}{e}\right) c_t(\mathsf{S}_t^{\star}),$$

which proves equation 15. □

The bound deserves a careful interpretation. Greedy selection provides a multiplicative approximation guarantee for the facility-location value F_t, not directly for the representation cost c_t. The two quantities differ by the constant shift bm_t, $F_t(\mathsf{S}) = bm_t - c_t(\mathsf{S})$, and this shift produces the additive term bm_t/e in equation 15. Therefore, the classical $(1 - 1/e)$ guarantee does not imply

$$c_t(\mathsf{S}_t^{\mathrm{G}}) \leq \alpha c_t(\mathsf{S}_t^{\star}),$$

for a universal multiplicative constant α.

This distinction is also why the convergence analysis in Section 6.2 is expressed using the actual representation cost attained by GhostMatch. That cost directly bounds the selection error through equation 5, whereas equation 15 provides a worst-case consequence of the facility-location approximation guarantee.

Stochastic greedy selection. Algorithm 2 replaces exact greedy selection with stochastic greedy selection Mirzasoleiman et al. (2015). Let $\varepsilon \in (0, 1)$ denote the stochastic-greedy accuracy parameter. Conditional on the candidate batch, the standard stochastic-greedy guarantee gives

$$\mathbb{E}_{\mathrm{sg}}\left[F_t(\mathsf{S}_t^{\mathrm{SG}})\right] \geq \left(1 - \frac{1}{e} - \varepsilon\right) F_t(\mathsf{S}_t^{\star}),$$

where $\mathbb{E}_{\mathrm{sg}}$ denotes expectation over the internal randomness of stochastic greedy. Since every returned subset has cardinality $k \geq 1$,

$$c_t(\mathsf{S}_t^{\mathrm{SG}}) = bm_t - F_t(\mathsf{S}_t^{\mathrm{SG}}).$$

Taking expectation and applying the preceding guarantee yields

$$\mathbb{E}_{\mathrm{sg}}\left[c_t(\mathsf{S}_t^{\mathrm{SG}})\right] \leq \left(\frac{1}{e} + \varepsilon\right) bm_t + \left(1 - \frac{1}{e} - \varepsilon\right) c_t(\mathsf{S}_t^{\star}). \tag{16}$$

Thus, Algorithm 2 can replace exact greedy selection without changing the convergence framework. Its approximation affects the representation cost realized by the selected subset, and that realized cost enters the convergence bound in exactly the same way.

D Non-Submodularity of the GREATS Selection Objective

We now show that the second-order Taylor surrogate for validation utility considered by GREATS Wang et al. (2024) is not submodular in general. Consequently, without additional structural assumptions, its greedy selection rule does not inherit the classical approximation guarantee for monotone submodular maximization.

Let B_t denote the candidate training batch and let z^{val} denote a validation example. For a subset $\mathsf{S} \subseteq \mathsf{B}_t$, consider the virtual update

$$\widetilde{\mathbf{w}}_{t+1}(\mathsf{S}) = \mathbf{w}_t - \eta_t \sum_{z_j \in \mathsf{S}} \mathbf{g}_{t,j},$$

where $\mathbf{g}_{t,j} = \nabla_{\mathbf{w}} \ell(\mathbf{w}_t, z_j)$. The improvement in validation loss associated with this virtual update is

$$U_t(\mathsf{S}) = \ell(\mathbf{w}_t, z^{\mathrm{val}}) - \ell(\widetilde{\mathbf{w}}_{t+1}(\mathsf{S}), z^{\mathrm{val}}).$$

Define $\mathbf{g}_t^{\mathrm{val}} = \nabla_{\mathbf{w}} \ell(\mathbf{w}_t, z^{\mathrm{val}})$, and $\mathbf{H}_t^{\mathrm{val}} = \nabla_{\mathbf{w}}^2 \ell(\mathbf{w}_t, z^{\mathrm{val}})$.

A second-order Taylor expansion of the validation loss around $\mathbf{w}_t$ gives the surrogate

$$\widetilde{U}_t(\mathsf{S}) = \eta_t \Big\langle \mathbf{g}_t^{\mathrm{val}}, \sum_{z_j \in \mathsf{S}} \mathbf{g}_{t,j} \Big\rangle - \frac{\eta_t^2}{2} \Big(\sum_{z_j \in \mathsf{S}} \mathbf{g}_{t,j} \Big)^\top \mathbf{H}_t^{\mathrm{val}} \Big(\sum_{z_j \in \mathsf{S}} \mathbf{g}_{t,j} \Big). \tag{17}$$

Proposition 3 *There exist a smooth loss function, a candidate batch of size two, and a validation example for which $\widetilde{U}_t(\mathsf{S})$ in equation 17 is not submodular. The conclusion continues to hold when $\mathbf{H}_t^{\mathrm{val}}$ is a positive scalar multiple of the identity.*

Proof. Let $\mathsf{B}_t = \{z_1, z_2\}$, and abbreviate $\mathbf{g}_{t,1}$ and $\mathbf{g}_{t,2}$ by $\mathbf{g}_1$ and $\mathbf{g}_2$. For any set function on a two-element ground set, submodularity requires

$$\widetilde{U}_t(\{z_1\}) + \widetilde{U}_t(\{z_2\}) \geq \widetilde{U}_t(\{z_1, z_2\}) + \widetilde{U}_t(\emptyset).$$

We first evaluate the difference between the two sides. Since $\widetilde{U}_t(\emptyset) = 0$, the linear validation-gradient terms cancel. Expanding the quadratic term for the two-element set gives

$$(\mathbf{g}_1 + \mathbf{g}_2)^\top \mathbf{H}_t^{\mathrm{val}} (\mathbf{g}_1 + \mathbf{g}_2) = \mathbf{g}_1^\top \mathbf{H}_t^{\mathrm{val}} \mathbf{g}_1 + \mathbf{g}_2^\top \mathbf{H}_t^{\mathrm{val}} \mathbf{g}_2 + 2\mathbf{g}_1^\top \mathbf{H}_t^{\mathrm{val}} \mathbf{g}_2,$$

where symmetry of the Hessian has been used. Therefore,

$$\widetilde{U}_t(\{z_1\}) + \widetilde{U}_t(\{z_2\}) - \widetilde{U}_t(\{z_1, z_2\}) - \widetilde{U}_t(\emptyset) = \eta_t^2 \mathbf{g}_1^\top \mathbf{H}_t^{\mathrm{val}} \mathbf{g}_2.$$

Thus, for this two-element ground set, submodularity would require $\mathbf{g}_1^\top \mathbf{H}_t^{\mathrm{val}} \mathbf{g}_2 \geq 0$.

There is no general reason for this condition to hold. A concrete one-dimensional example establishes the claim. Let the parameter be $w \in \mathbb{R}$ and take the current iterate to be $w_t = 0$. Define the two candidate losses by

$$\ell(w, z_1) = \frac{1}{2}(w+1)^2, \qquad \ell(w, z_2) = \frac{1}{2}(w-1)^2.$$

Their gradients at $w_t = 0$ are $g_1 = 1$, and $g_2 = -1$. Choose the validation loss

$$\ell(w, z^{\mathrm{val}}) = \frac{h}{2} w^2, \qquad h > 0.$$

At $w_t = 0$, its Hessian is $H_t^{\mathrm{val}} = h > 0$. Hence, $g_1 H_t^{\mathrm{val}} g_2 = -h < 0$.

The submodularity difference is therefore

$$\widetilde{U}_t(\{z_1\}) + \widetilde{U}_t(\{z_2\}) - \widetilde{U}_t(\{z_1, z_2\}) - \widetilde{U}_t(\emptyset) = -\eta_t^2 h < 0.$$

This directly violates the required submodularity inequality.

The example also satisfies the stronger statement in the proposition. In one dimension, $H_t^{\mathrm{val}} = h$ with $h > 0$ is simply a positive scalar multiple of the identity. if

$$\mathbf{H}_t^{\mathrm{val}} = \lambda \mathbf{I}, \qquad \lambda > 0,$$

then $\mathbf{g}_1^\top \mathbf{H}_t^{\mathrm{val}} \mathbf{g}_2 = \lambda \mathbf{g}_1^\top \mathbf{g}_2$. Any negatively aligned pair of candidate gradients therefore makes the submodularity difference negative. Thus, $\widetilde{U}_t$ is not submodular in general. □

Proposition 3 does not imply that greedy selection is ineffective for GREATS. It establishes only that the Taylor surrogate in equation 17 does not generally satisfy the structural condition required to invoke the classical $(1 - 1/e)$ approximation guarantee for monotone submodular maximization Nemhauser et al. (1978). GhostMatch differs in this respect because its facility-location objective is monotone and submodular by construction.

E Additional Experimental Details

E.1 Experimental Configurations

Table 2 summarizes the model, fine-tuning dataset, and evaluation dataset used in each of the four experimental configurations reported in Section 7.

Table 2: Model, fine-tuning dataset, and evaluation dataset for the four experimental configurations.

Model	Fine-Tuning Dataset	Evaluation Dataset
TinyLlama-1.1B (Zhang et al., 2024)	LESS (Xia et al., 2024)	MMLU (Hendrycks et al., 2020)
Llama-2-7B (Touvron et al., 2023)	LESS	MMLU
Mistral-7B (Jiang et al., 2023)	LESS	TyDiQA (Clark et al., 2020)
Llama-3-8B (AI@Meta, 2024)	Alpaca (Taori et al., 2023)	SamSum (Gliwa et al., 2019)

E.2 Full Exactness Ablation Results

Section 7.3 isolates the effect of gradient fidelity by comparing GhostMatch with ZeroOrder. The two methods use the same facility-location selection rule but construct its gradient geometry from different signals: GhostMatch uses exact ghost-computed gradient inner products, whereas ZeroOrder uses a zeroth-order approximation. Table 3 reports the corresponding configuration-level results.

Table 3: GhostMatch versus ZeroOrder on the primary metric of each configuration. The margin is oriented so that a positive value always favors GhostMatch.

	TinyLlama-1.1B MMLU ppl	Llama-2-7B MMLU ppl	Mistral-7B TyDiQA F1	Llama-3-8B SamSum R1
GhostMatch	5.99	9.73	0.609	0.090
ZeroOrder	6.15	9.95	0.585	0.036
Margin	$+0.16$	$+0.22$	$+0.024$	$+0.054$

The advantage is also consistent at the individual-run level rather than arising only after averaging across seeds. As shown in Table 4, GhostMatch outperforms ZeroOrder on the corresponding primary metric in every matched final evaluation: $6/6$ runs for TinyLlama-1.1B, $6/6$ for Llama-2-7B, $2/2$ for Mistral-7B, and $2/2$ for Llama-3-8B, for a total of $16/16$ matched runs. This provides additional evidence that the improvement observed in Section 7.3 is consistent across the evaluated configurations rather than being driven by a small number of seeds.

Table 4: Seed-level consistency of the exactness ablation. A win indicates that GhostMatch achieves a better final value than ZeroOrder on the primary metric used for that configuration.

Configuration	Matched Runs	GhostMatch Wins
TinyLlama-1.1B / MMLU	6	$6/6$
Llama-2-7B / MMLU	6	$6/6$
Mistral-7B / TyDiQA	2	$2/2$
Llama-3-8B / SamSum	2	$2/2$
Total	16	$16/16$

E.3 Additional Evaluation Metrics

The main experiments in Section 7 report one primary metric for each configuration. For completeness, we also report complementary metrics from the same evaluation runs to assess whether the main conclusions are sensitive to the choice of primary metric.

For the two MMLU configurations, Table 5 reports classification accuracy in addition to the answer perplexity used in the main results. On Llama-2-7B, GhostMatch attains the highest mean MMLU accuracy among the five methods despite using no held-out validation data.

Table 6: Additional SamSum evaluation metrics for the Llama-3-8B configuration.

Method	ROUGE-1	ROUGE-2	ROUGE-L
Regular	0.1740	0.0570	0.1321
GREATS	0.1324	0.0377	0.1002
GhostMatch	0.0902	0.0231	0.0680
OPUS	0.1562	0.0486	0.1181
ZeroOrder	0.0358	0.0092	0.0268

Table 5: MMLU accuracy (%) for the two MMLU configurations. Values are mean $\pm$ standard deviation over the corresponding evaluation runs.

Method	TinyLlama-1.1B	Llama-2-7B
Regular	23.33 ± 1.15	56.67 ± 2.31
GREATS	24.67 ± 0.58	56.33 ± 0.58
GhostMatch	23.33 ± 0.58	57.67 ± 1.15
OPUS	25.00 ± 0.00	56.00 ± 0.00
ZeroOrder	23.00 ± 0.00	56.33 ± 0.58

Thus, at the 300-step horizon used in the main experiments, the Llama-3-8B configuration remains a cold-start regime in which subset selection does not yet outperform full-batch training, and the exact-gradient ablation is not specific to the choice of ROUGE-1.

E.4 Exactness Advantage Throughout Training

The endpoint advantage in Table 4 is also visible throughout training. Figure 2 plots the paired performance difference between GhostMatch and ZeroOrder at every saved nonzero evaluation checkpoint. For TinyLlama-1.1B, the comparison includes two MMLU subjects and three seeds per subject, giving six matched trajectories and five checkpoints per trajectory. GhostMatch achieves lower test perplexity in 29 of the 30 paired checkpoint comparisons. For Mistral-7B on TyDiQA, GhostMatch achieves higher F1 in all 10 paired comparisons across two seeds and five checkpoints. Overall, 39 of 40 available checkpoint-level comparisons favor GhostMatch.

The single exception occurs at the first nonzero TinyLlama checkpoint for one subject–seed pair; all subsequent comparisons for that run favor GhostMatch. Thus, the final improvements in Table 3 generally reflect an advantage that persists during optimization rather than one that appears only at the final checkpoint.

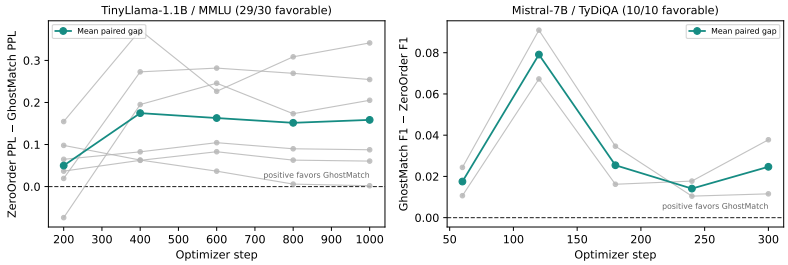


Figure 2: Exact-gradient advantage throughout training. Each thin curve corresponds to a matched subject–seed pair for TinyLlama-1.1B or a matched seed for Mistral-7B, and the thick curve shows the mean paired gap. For TinyLlama, the gap is ZeroOrder test perplexity minus GhostMatch test perplexity; for Mistral, it is GhostMatch F1 minus ZeroOrder F1. Positive values therefore favor GhostMatch in both panels. Excluding the shared step-0 initialization, GhostMatch is better in $29/30$ TinyLlama comparisons and $10/10$ Mistral comparisons, for $39/40$ paired checkpoint comparisons overall.

F EXTENDED-HORIZON ROBUSTNESS CHECKS

Table 1 reports performance at the training horizons used throughout the main experiments, namely 1000 optimizer steps for the MMLU settings and 300 steps for TyDiQA and SamSum. These budgets provide the comparisons used in the main text, but they leave open whether the observed differences are specific to those horizons or persist under longer training. This question is particularly relevant for the cold-start setting in Section 7.5, where training on the full candidate batch outperforms every subset-selection method at step 300. We therefore extend training to 3000 steps for MMLU and TyDiQA and to 1000 steps for SamSum, while leaving the remaining hyperparameters unchanged. The MMLU experiments retain the three-seed protocol used in the main experiments, while the extended TyDiQA and SamSum experiments use three seeds rather than the two reported in Table 1. Because these experiments contain only three seeds, small differences between closely clustered methods should be interpreted qualitatively rather than as statistically significant rankings.

F.1 MMLU UNDER LONGER TRAINING

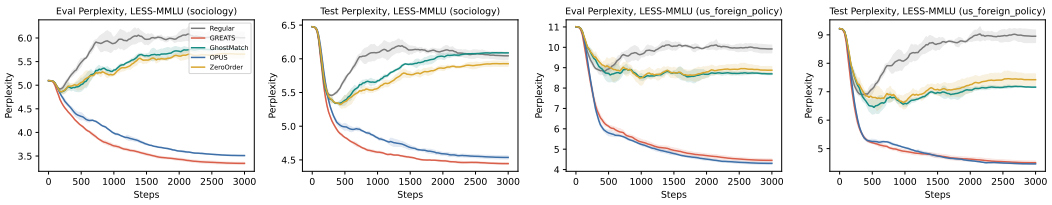


Figure 3: Extended-horizon MMLU trajectories at learning rate 2×10^{-5}, continued to 3000 optimizer steps. Curves show the mean over three seeds and shaded regions show one standard deviation. The first two panels report evaluation and test perplexity on sociology, and the final two panels report the corresponding quantities on U.S. foreign policy.

At the learning rate used in the main comparison, the broad MMLU ordering remains stable under substantially longer training. At step 3000, averaging test perplexity across sociology and U.S. foreign policy gives GREATS at 4.47, OPUS at 4.50, GhostMatch at 6.62, ZeroOrder at 6.68, and Regular at 7.50. Thus, the same ordering observed at the shorter horizon persists after three times as many optimizer steps. In particular, the validation-directed methods retain their advantage, while GhostMatch continues to outperform both ZeroOrder and full-candidate-batch training at this learning rate.

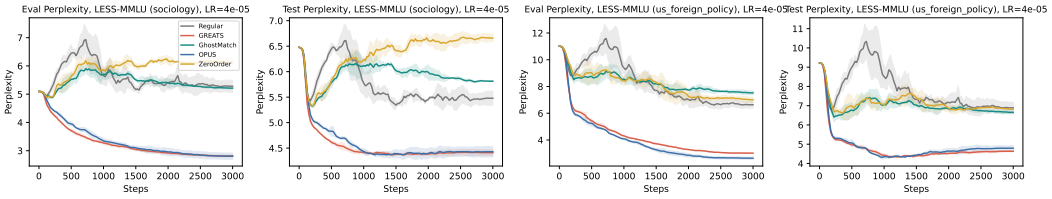


Figure 4: Extended-horizon MMLU trajectories at learning rate 4×10^{-5}, continued to 3000 optimizer steps. Curves show the mean over three seeds and shaded regions show one standard deviation.

Figure 4 repeats the extended experiment after doubling the learning rate to 4×10^{-5}. GREATS and OPUS remain the two strongest methods, with average test perplexities of 4.52 and 4.61, respectively. Regular and GhostMatch are closely grouped at 6.18 and 6.23, followed by ZeroOrder at 6.74. The exact ordering among the validation-free methods therefore depends somewhat on the learning rate, but the main qualitative conclusion is unchanged. Longer training does not erase the separation between the validation-directed methods and the remaining methods, and the extended trajectories do not indicate that the main MMLU comparison is an artifact of stopping at 1000 steps.

F.2 TyDiQA Under Longer Training

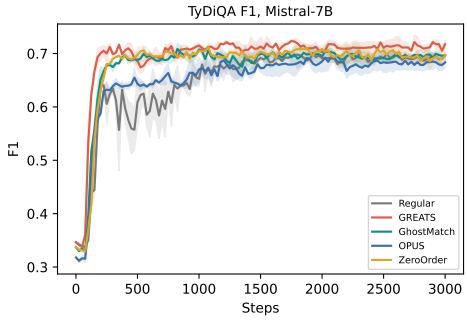


Figure 5: TyDiQA F1 on Mistral-7B extended from 300 to 3000 optimizer steps. Curves show the mean over three seeds and shaded regions show one standard deviation.

The TyDiQA trajectories show a different effect. At the 300-step horizon used in the main text, performance spans approximately 0.14 F1 across methods. By step 3000, the spread has narrowed to approximately 0.034. GREATS remains highest at 0.717, while ZeroOrder, Regular, and GhostMatch finish at 0.697, 0.696, and 0.696, respectively, and OPUS finishes at 0.683. The principal extended-horizon observation is therefore the compression of the performance gaps rather than the precise ordering inside the final cluster.

This result provides an important qualification to the shorter-horizon comparison. The differences observed at step 300 are meaningful at that training budget, but several of them become substantially smaller as optimization continues. We do not interpret this downstream F1 behavior as a direct test of the SGD convergence bound in Section 6, particularly because the experiments use an adaptive optimizer. Rather, the extended run shows empirically that the relative advantage of subset selection can depend strongly on the training horizon.

F.3 SamSum and the Cold-Start Effect

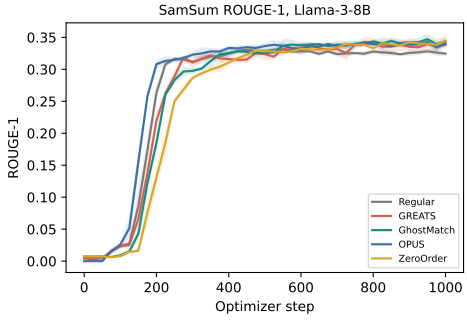


Figure 6: SamSum ROUGE-1 on Llama-3-8B extended from 300 to 1000 optimizer steps. Curves show the mean over three seeds and shaded regions show one standard deviation.

The extended SamSum experiment directly revisits the boundary case in Section 7.5. At step 300, Regular reaches 0.174 ROUGE-1 and outperforms every method that trains on a selected subset, while GhostMatch reaches 0.090. The longer trajectory shows that this advantage does not persist. At step 1000, ZeroOrder reaches 0.345, GhostMatch 0.343, OPUS 0.340, GREATS 0.338, and Regular 0.324. All five methods now lie within approximately 0.02 ROUGE-1, compared with a spread of approximately 0.14 at the original horizon.

This behavior strengthens the cold-start interpretation in Section 7.5. The short-horizon advantage of using the entire candidate batch is real at the training budget reported in the main text, but it is not a persistent disadvantage of subset selection. Once training progresses beyond the early phase,

Regular loses its initial advantage and the subset-selection methods catch up. The trajectories remain closely grouped near the end of the extended run, so we do not treat their exact ordering at step 1000 as an asymptotic ranking. The more robust conclusion is that the failure mode observed at step 300 is transient.

F.4 SUMMARY

The extended-horizon experiments provide a robustness check on the training budgets used in Table 1. On MMLU, the main ordering at the primary learning rate persists through 3000 steps, while an additional learning-rate experiment preserves the broader separation between the validation-directed and validation-free methods. On TyDiQA, the initially larger performance gaps compress substantially under longer training. On SamSum, the early advantage of full-candidate- batch training disappears, supporting the cold-start interpretation in Section 7.5. These experiments therefore complement rather than replace the main results. The shorter horizons remain the relevant comparison at the training budgets studied in the main text, while the extended runs show which conclusions remain stable and which are specific to an earlier stage of optimization.