Holdout-Loss-Based Data Selection for LLM Finetuning via In-Context Learning

Ling Zhang*, Xianliang Yang*, Juwon Yu, Park Cheonyoung, Lei Song, Jiang Bian

Microsoft Research Asia · Korean KT

*Equal contribution

Why Data Quality Matters

Fine-tuning drives model alignment

- Supervised fine-tuning (SFT) and preference optimization (DPO, SimPO)
- Model behavior strongly depends on training examples

But training data is noisy

- 20–40% of preference pairs may be mislabeled
- Noisy supervision directly degrades alignment quality

Less data, if curated, can match much larger sets

- Careful selection matters more than dataset size

Which training examples are actually worth learning from?

Problem Formulation

Large noisy training set $\mathcal{D} = \{(x_i, y_i)\}$
Small high-quality holdout set $\mathcal{D}_{ho} = \{(x_i^{ho}, y_i^{ho})\}$
Goal: select subset that minimizes holdout loss

$$\bar{\mathcal{D}}^* = \arg\min_{\bar{\mathcal{D}} \subseteq \mathcal{D}} \mathcal{L}\big(\mathcal{D}_{ho};\, \theta^*(\bar{\mathcal{D}})\big)$$

Solving this directly requires retraining on every subset — intractable.

→ **Instead: assign each example a score reflecting its contribution to $\mathcal{L}(\mathcal{D}_{ho})$, then reweight gradient updates accordingly.**

Existing Approaches

Heuristics

LIMA, AlpaGasus, IFD

- Proxy signals: length, perplexity, reward
- No theoretical grounding
- Often task-specific

Influence Functions

LESS, GREATS

- Estimate loss change via Taylor expansion
- Requires gradient / Hessian storage
- Expensive for large models

Existing Approaches

RHO-Loss (Mindermann et al., 2022)

- Scores each example by $\ell(y|x;\,\theta_t) - \ell(y|x;\,\theta^*(\mathcal{D}_{ho}))$ — the reducible holdout loss
- Requires training a **separate reference model** on $\mathcal{D}_{ho}$
- Fixed reference introduces **bias** as the main model evolves

Goal: principled scoring — without extra model training

Key Idea: In-Context Approximation (ICA)

Observation (Dai et al., 2023) — In-context learning implicitly performs gradient descent: conditioning on examples approximates explicit parameter updates.

Applying Bayes' rule (Mindermann et al., 2022) rewrites the holdout loss after adding (x, y) as:

$$\mathcal{L}(\mathcal{D}_{ho};\, \theta^*(\mathcal{D}_t \cup \{(x,y)\})) = \ell(y \mid x;\, \theta^*(\mathcal{D}_t \cup \mathcal{D}_{ho})) - \ell(y \mid x;\, \theta_t) - \mathcal{L}(\mathcal{D}_{ho};\, \theta_t)$$

Instead of retraining on $\mathcal{D}_t \cup \mathcal{D}_{ho}$ to evalutate the first term, we approximate it by **conditioning on $\mathcal{D}_{ho}$ in context**:

$$\ell(y \mid x;\ \theta^*(\mathcal{D}_t \cup \mathcal{D}_{ho})) \ \approx\ \ell(y \mid x, \mathcal{D}_{ho};\ \theta_t)$$

Key Idea: In-Context Approximation (ICA)

Omitting terms independent of (x, y), ICA score is

$$\boxed{s_{\mathrm{ICA}}(x, y; \theta_t) \;=\; \ell(y \mid x;\, \theta_t) \;-\; \ell(y \mid x, \mathcal{D}_{ho};\, \theta_t)}$$

— no auxiliary model, updates dynamically with θ_t

High score → example reduces holdout loss → upweight it

Method: ICA Score-Based Reweighting

Step 1 — Compute ICA score for each $(x_i, y_i) \in \mathcal{B}_t$

$$s_i = \ell(y_i \mid x_i;\, \theta_t) - \ell(y_i \mid x_i, \mathcal{D}_{ho};\, \theta_t)$$

Step 2 — Normalize to weights (min–max)

$$w_i = \frac{s_i - \min_j s_j}{\max_j s_j - \min_j s_j} \in [0, 1]$$

Step 3 — Weighted gradient update

$$g_t = \sum_{i=1}^{|\mathcal{B}_t|} w_i \, \nabla_\theta \, \ell(x_i, y_i;\, \theta_t)$$

Practical optimizations

kNN retrieval — use top $k = 3$ similar holdout examples per candidate, not the full $\mathcal{D}_{ho}$

Periodic score updates — recompute scores R times total

Results

Win Rate vs. Baselines

Setting	vs. Standard Training	vs. RHO-Loss	vs. One-Shot
SFT	**71–85%**	47–54%	57–67%
DPO	**61–80%**	47–57%	54–60%
SimPO	**62–94%**	45–51%	49–57%

Models: LLaMA 3B/8B, Qwen 4B/8B · Win rate judged by GPT-4o

Compute Overhead

Method	Extra Overhead
ICA (ours)	**~1.5%**
One-Shot	~4%
RHO-Loss	~10%

Results

Key Observations

- Consistently beats standard training across **all paradigms and model families**
- Matches or exceeds RHO-Loss **— without a reference model**
- Competitive with LESS / GREATS on SFT at substantially **lower cost**
- Effective noise mitigation on reasoning tasks (GSM8K)

Takeaway: better alignment at negligible compute overhead

Conclusion

Core Idea

Estimate each training example's contribution to holdout loss via **in-context approximation** — no auxiliary model, no retraining

$$s_{\text{ICA}}(x, y; \theta_t) = \ell(y \mid x;\, \theta_t) - \ell(y \mid x, \mathcal{D}_{ho};\, \theta_t)$$

→ Use scores to **dynamically reweight gradient updates** during fine-tuning

What This Shows

- In-context approximation can replace explicit influence estimation
- Simple scoring improves alignment across **SFT, DPO, and SimPO**
- Practical for modern LLM pipelines: only **~1.5% compute overhead**

Conclusion

Limitations

- Score quality depends on the **representativeness of** $\mathcal{D}_{ho}$ — a noisy holdout degrades scores
- On-policy methods (e.g., PPO) require frequent re-scoring of newly generated data — **not yet addressed**

Future Directions

- **Adaptive holdout construction —** automatically refine $\mathcal{D}_{ho}$ during training
- Extending ICA to **on-policy RL** (PPO, GRPO)
- Combining ICA scores with **reward-model signals**

Code: `github.com/microsoft/HeurAgenix/tree/dpo`