Test-Time Training for Zero-Resource Dense Retrieval Reranking

Shiyan Liu[1] **Yichen Li**[2]
[1]Huazhong University of Science and Technology [2]ByteDance
shyl@hust.edu.cn liyichen.1@bytedance.com

Abstract

Dense retrievers excel at first-stage candidate generation but lack effective reranking in zero-resource settings. Existing approaches face a fundamental dilemma: cross-encoders deliver strong reranking quality but require costly supervised training and incur high latency, while unsupervised BM25 reranking consistently degrades dense retrieval performance on most of BEIR benchmarks. We propose **DART** (**D**ense **A**daptive **R**eranking at **T**est-time), which resolves this dilemma by adapting the scoring function at inference time. For each query, the top-ranked documents serve as pseudo-positive examples and the bottom-ranked as pseudo-negative examples, providing noisy but readily available supervision to adapt a bilinear scoring matrix W via a small number of gradient updates. We further introduce a confidence-weighted margin loss and a cross-query momentum buffer that warm-starts adaptation across queries. On six BEIR benchmarks, DART achieves a mean per-dataset relative NDCG@10 gain of **+2.1%** over the dense retrieval baseline with under 10ms additional latency per query, demonstrating a powerful capability for zero-shot performance enhancement and cross-domain generalization.

1 Introduction

The modern information retrieval pipeline is typically organized as a two-stage cascade: a fast first-stage retriever narrows the corpus to a candidate set, which is then reranked by a more precise but computationally expensive model (Lin et al., 2022; Guo et al., 2020). Bi-encoder dense retrievers (Karpukhin et al., 2020; Reimers and Gurevych, 2019) have become the standard first stage, offering strong recall with sub-millisecond per-document scoring. However, reranking remains an open problem in *zero-resource* deployments where no labeled relevance judgments exist for the target domain.

Supervised rerankers address this with extra training. Cross-encoders (Nogueira and Cho, 2019; Nogueira et al., 2020) jointly attend to query and document, achieving high accuracy at the cost of 200–500ms latency and substantial labeled data. Recent LLM-based rerankers (Sun et al., 2023; Weller et al., 2025) push accuracy further but amplify both requirements. In the absence of training data, practitioners typically fall back to the dense retrieval ranking itself—forgoing any reranking step entirely—because no lightweight, reliable alternative exists. This is especially true in deployments built entirely around vector databases (Johnson et al., 2019), where only dense embeddings are indexed and lexical systems such as BM25 (Robertson and Zaragoza, 2009) are not available.

We observe that a useful supervision signal is already present at inference time, without any external resource: the ranked list produced by the dense retriever itself. The top-ranked documents for a given query are likely relevant; the bottom-ranked are likely not. Although this pseudo-labeling is noisy, it captures query-specific relevance structure that the fixed, query-agnostic cosine scoring function cannot exploit. This motivates a *Test-Time Training* (TTT) approach (Sun et al., 2020; Liu et al., 2021): rather than modifying query or document representations, we adapt the *scoring function* directly for each incoming query using only its own retrieved candidates as supervision.

We propose **DART**, which frames reranking as a per-query optimization problem. Given a query, we initialize a bilinear scoring matrix W to the identity and perform a small number of gradient steps using a confidence-weighted margin loss over pseudo-labeled positives and negatives drawn from the top-K retrieved documents. We additionally introduce a cross-query momentum buffer that accumulates adaptation signals across the query stream to warm-start each new query, and a dataset-adaptive optimizer selection strategy that balances convergence speed against pseudo-label noise. Evaluated on six BEIR benchmarks (Thakur et al., 2021), DART

Proceedings of the 4th Workshop on Towards Knowledgeable Foundation Models (KnowFM 2026), pages 105–114
July 3, 2026 ©2026 Association for Computational Linguistics

achieves a mean per-dataset relative NDCG@10 gain of +2.1% over the dense retrieval baseline with under 10ms latency per query.

Our contributions are summarized as follows:

- We propose DART, a principled TTT framework for zero-resource dense retrieval reranking that adapts a bilinear scoring matrix at inference time using pseudo-labels derived directly from the dense retrieval ranking, requiring no external resource.
- We empirically demonstrate that DART improves over the dense retrieval baseline on five of six BEIR benchmarks with a mean per-dataset relative NDCG@10 gain of $+2.1\%$ and under 10ms additional latency per query.
- We provide interpretability analysis showing that W updates concentrate in a low-dimensional subspace correlated with query difficulty, providing empirical evidence for the structural basis of cross-domain generalization.

2 Related Work

2.1 Neural Reranking

Neural reranking has evolved through three generations. Early cross-encoder models (Nogueira and Cho, 2019) apply BERT (Devlin et al., 2019) to jointly encode query-document pairs, achieving strong performance at the cost of high latency. MonoT5 (Nogueira et al., 2020) reformulates reranking as a sequence-to-sequence generation task. ColBERTv2 (Santhanam et al., 2022) introduces late interaction to balance effectiveness and efficiency. More recently, listwise Large Language Model (LLM) rerankers (Sun et al., 2023; Pradeep et al., 2023) leverage the in-context learning capabilities of LLMs. Weller et al. (2025) train rerankers on reasoning traces from DeepSeek-R1 (DeepSeek-AI et al., 2025), achieving state-of-the-art performance by exploiting test-time compute in the form of chain-of-thought reasoning—a complementary direction to ours, which targets lightweight parameter adaptation rather than extended generation. All supervised rerankers require labeled training data, limiting applicability in zero-resource domains.

2.2 Unsupervised Domain Adaptation

GPL (Wang et al., 2022a) generates pseudo training pairs using a cross-encoder teacher for unsupervised domain adaptation, but still requires offline training. AugTriever (Zhuang et al., 2023) constructs pseudo query-document pairs via query extraction and generation for unsupervised retrieval pretraining. Meng et al. (2022) propose relevance-aware contrastive pretraining that weights pseudo-positive pairs by estimated relevance, improving Contriever (Izacard et al., 2021) on BEIR (Thakur et al., 2021) without labeled data. UDAPDR (Saad-Falcon et al., 2023) uses LLMs to generate domain-specific queries for zero-shot dense retrieval adaptation. These methods improve the retrieval model itself through data augmentation and pretraining; DART instead adapts the *scoring function* at inference time with no offline training.

2.3 Pseudo Relevance Feedback

Pseudo Relevance Feedback (PRF) (Lavrenko and Croft, 2017) assumes the top-k retrieved documents are relevant and uses them to expand queries. Dense PRF methods (Li et al., 2023) encode feedback documents and aggregate their embeddings with the query embedding. ColBERT-PRF (Wang et al., 2023) applies late interaction with pseudo-relevant embeddings. PromptPRF (Li et al., 2025) uses LLMs to extract structured features from top-ranked documents offline, enabling small retrievers to match larger ones. Wang et al. (2022b) show that dense retrievers benefit from interpolation with BM25, motivating our hybrid pseudo-label strategy. Unlike PRF methods that modify query representations, DART modifies the *scoring matrix*—a distinct and complementary approach that preserves the original query and document embeddings.

2.4 Test-Time Training

TTT (Sun et al., 2020) adapts model parameters at inference time using self-supervised signals from the test input. TTT++ (Liu et al., 2021) improves stability through feature alignment. LoRA-based TTT (Yu et al., 2023) and related work on TTT for abstract reasoning (Akyürek et al., 2024) demonstrate that even a handful of gradient steps on a test instance can substantially improve performance. In information retrieval, TTT has not been studied as a reranking mechanism. The closest work is Weller et al. (2025), which applies test-time *compute scaling* rather than test-time *parameter adaptation*. DART is among the first methods to explore parameter-level TTT specifically for retrieval reranking.

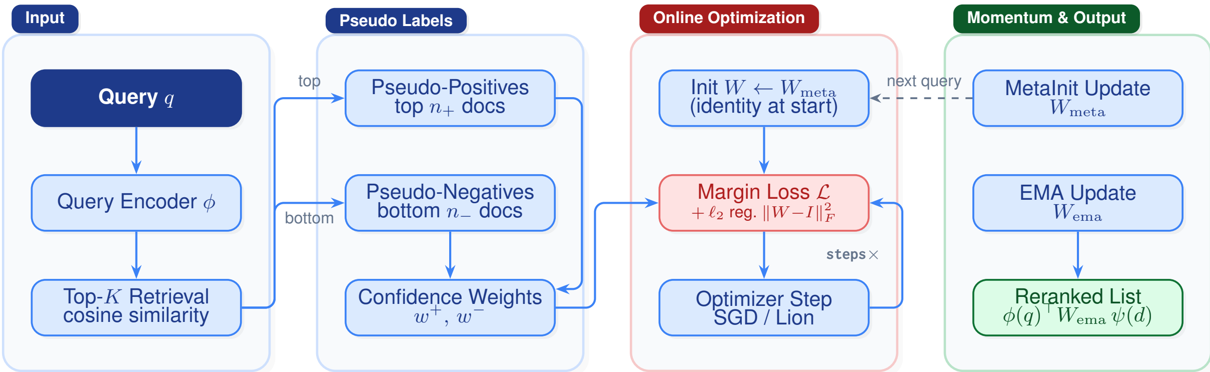

Figure 1: Overall DART algorithm flowchart

3 Method

3.1 Setup

Let $\phi : \mathcal{Q} \to \mathbb{R}^d$ and $\psi : \mathcal{D} \to \mathbb{R}^d$ be fixed, pretrained query and document encoders (e.g., a sentence transformer). Standard dense retrieval scores a query-document pair by cosine similarity after ℓ_2-normalization:

$$s(q, d) = \phi(q)^\top \psi(d). \tag{1}$$

This scoring function implicitly treats all embedding dimensions as equally important and independent. However, for a specific query, certain semantic dimensions are more discriminative than others. For example, a query about "cardiovascular disease prevention" should upweight dimensions encoding health interventions and downweight dimensions encoding economic or political concepts.

To enable query-specific reweighting of embedding dimensions, we generalize the scoring function to a bilinear form:

$$s_W(q, d) = \phi(q)^\top W \psi(d), \tag{2}$$

where $W \in \mathbb{R}^{d \times d}$ is a transformation matrix. Decomposing $W = I + \Delta W$ reveals that the adjustment ΔW serves as a query-specific correction to the identity mapping. Initializing $W = I$ recovers the standard cosine score exactly, providing a natural starting point and a reliable baseline.

The core challenge is to estimate ΔW for each query at inference time, using only the retrieved documents as a source of noisy supervision, without any labeled data.

3.2 Overall Framework

We treat the reranking task as an online optimization problem. For an incoming query q, we first retrieve its top-K documents using the initial scoring function $s(q, d)$. These top documents, albeit noisy, provide a set of pseudo-positive and pseudo-negative examples for adaptation (Lavrenko and Croft, 2017). We then update W by minimizing a loss function designed to pull relevant documents closer and push irrelevant ones away, while regularizing W towards the identity to avoid overfitting.

To improve the robustness and convergence of the online update, we introduce three components that mirror a standard optimization pipeline:

1. **Learning objective** (Section 3.3): a loss with confidence-weighted pseudo labels and an adaptive margin.

2. **Cross-query momentum** (Section 3.4): temporal regularizers (MetaInit and EMA) that transfer knowledge across queries.

3. **Optimizer selection** (Section 3.5): a dataset-driven selection between a conservative optimizer (Stochastic Gradient Descent, SGD) and a more aggressive one (Lion (Chen et al., 2023)).

The overall algorithm is illustrated in Figure 1.

3.3 Learning Objective

For a given query q, let the initial retrieval scores of the top-K documents be $s_1 \geq s_2 \geq \cdots \geq s_K$. We treat the top n_{pos} documents as pseudo-positive and the bottom n_{neg} as pseudo-negative.

Confidence-weighted pseudo labels. To reduce the impact of label noise, we assign soft weights to the pseudo-positive and pseudo-negative examples. Define the normalization constant for the positive set:

$$Z_{\text{pos}} = \sum_{j=1}^{n_{\text{pos}}} \exp(s_j / T), \tag{3}$$

where $T > 0$ is a temperature hyperparameter that controls the concentration of weights. The weight of the i-th pseudo-positive document is then:

$$w_i^+ = \frac{\exp(s_i/T)}{Z_{\text{pos}}}, \qquad i = 1, \ldots, n_{\text{pos}}. \quad (4)$$

Similarly, for the pseudo-negative documents, we define:

$$Z_{\text{neg}} = \sum_{k=K-n_{\text{neg}}+1}^{K} \exp(-s_k/T), \quad (5)$$

and the weight of the j-th pseudo-negative document is:

$$w_j^- = \frac{\exp(-s_j/T)}{Z_{\text{neg}}}, \qquad j = K-n_{\text{neg}}+1, \ldots, K. \quad (6)$$

These weights assign higher importance to examples with larger initial scores (for positives) or more negative scores (for negatives), effectively focusing the learning on high-confidence pseudo-labels.

Adaptive margin (AdaMargin). The loss function encourages a margin between the aggregated scores of pseudo-positive and pseudo-negative documents. Because queries vary in difficulty, we make the margin adaptive to the highest initial similarity $s_{\text{top1}} = s_1$:

$$\text{margin}(q) = \alpha_{\text{mar}} + \beta_{\text{mar}}\,(1 - s_{\text{top1}}), \quad (7)$$

where α_{mar} and β_{mar} are hyperparameters that determine the base margin and the strength of the difficulty-based adaptation.

Loss. Recall the bilinear scoring function $s_W(q, d) = \phi(q)^\top W \psi(d)$ (Section 3.1). We first compute the total weighted score for the pseudo-positive documents:

$$P = \sum_{i=1}^{n_{\text{pos}}} w_i^+\, s_W(q, d_i). \quad (8)$$

Similarly, the total weighted score for the pseudo-negative documents is:

$$N = \sum_{j=K-n_{\text{neg}}+1}^{K} w_j^-\, s_W(q, d_j). \quad (9)$$

The ranking loss is then defined as:

$$\mathcal{L}_{\text{rank}} = \max\Big(0,\ \text{margin}(q) - P + N\Big). \quad (10)$$

We add an ℓ_2 regularization term to keep W close to the identity:

$$\mathcal{L}_{\text{reg}} = \lambda \|W - I\|_F^2, \quad (11)$$

where $\lambda > 0$ is a hyperparameter.

The total loss for the current query is:

$$\mathcal{L}(W) = \mathcal{L}_{\text{rank}} + \mathcal{L}_{\text{reg}}. \quad (12)$$

3.4 Cross-Query Momentum

To transfer knowledge across queries and smooth the parameter evolution, we maintain two complementary momentum-like states. Let t denote the index of the current query. The transformation matrix after updating query t is denoted $W_{\text{star}}^{(t)}$.

Meta Initialization (MetaInit). We learn a global initial matrix W_{meta} that is passed from one query to the next. Before updating query t, the initial matrix is set to the meta parameter obtained after processing the previous query:

$$W_{\text{init}}^{(t)} = W_{\text{meta}}^{(t-1)}.$$

After obtaining $W_{\text{star}}^{(t)}$, we update the meta parameter using the Reptile rule:

$$W_{\text{meta}}^{(t)} = W_{\text{meta}}^{(t-1)} + \beta_{\text{meta}} \left(W_{\text{star}}^{(t)} - W_{\text{meta}}^{(t-1)}\right), \quad (13)$$

where $\beta_{\text{meta}} > 0$ is a meta learning rate. This provides an increasingly better starting point for each new query, accelerating adaptation over time.

Exponential Moving Average (EMA). We maintain an exponentially decaying average of the transformation matrices for stability:

$$W_{\text{ema}}^{(t)} = \alpha_{\text{ema}}\, W_{\text{ema}}^{(t-1)} + (1 - \alpha_{\text{ema}})\, W_{\text{star}}^{(t)}, \quad (14)$$

with $\alpha_{\text{ema}} \in (0, 1)$ a decay hyperparameter. The smoothed matrix $W_{\text{ema}}^{(t)}$ is used for re-ranking the current query, which reduces the variance of the updates.

Both states are carried over across the query stream. MetaInit affects the initial value of the next query, while EMA smooths the output of the current query.

3.5 Optimizer Selection

The choice of optimizer directly affects how each query's loss is minimized and interacts with the cross-query states. Based on empirical observations across diverse datasets, we provide guidelines for selecting between two optimizers.

SGD with momentum. SGD with momentum ($\mu = 0.9$) performs conservative updates:

$$v_{t+1} = \mu v_t - \eta \nabla \mathcal{L}(W_t), \quad W_{t+1} = W_t + v_{t+1}, \tag{15}$$

where η is the learning rate. This optimizer is preferable when the initial dense retrieval is noisy or the dataset suffers from high pseudo-label uncertainty (e.g., TREC-COVID, SciFact), as it avoids overfitting.

Lion optimizer. The Lion optimizer updates parameters using only the sign of the gradient:

$$W_{t+1} = W_t - \eta \cdot \text{sign}\big(\beta_1 m_t + (1-\beta_1)\nabla \mathcal{L}(W_t)\big), \tag{16}$$

with m_t an exponential moving average of past gradients. Lion discards gradient magnitude, making it robust to scale variations and often faster to converge. It is more suitable for datasets where dense retrieval already provides clean pseudo-labels (e.g., NFCorpus, FiQA, SCIDOCS, ArguAna (Thakur et al., 2021)).

Practice. When no prior knowledge about the dataset is available, we recommend a simple warm-up adaptive strategy: process the first 50–100 queries with both optimizers, compare their average pseudo-label loss (Section 3.3), and select the optimizer with the lower loss for the remaining queries. This adds negligible overhead and eliminates manual tuning. In our experiments, we report the better result for each dataset following this rule or the empirical guidelines above.

Pseudo-code. Algorithm 1 summarizes the complete test-time adaptation for a single query.

Algorithm 1 DART for One Query

Require: Query q; encoders ϕ, ψ; retrieval depth K; hyperparameters $n_{\text{pos}}, n_{\text{neg}}, T, \alpha_{\text{mar}}, \beta_{\text{mar}}, \lambda, \alpha_{\text{ema}}, \beta_{\text{meta}}$, steps, η; optimizer (SGD / Lion)
Ensure: Reranked list of K documents
Global state: $W_{\text{meta}}, W_{\text{ema}}$ (initialized to I)

$\{(d_k, s_k)\}_{k=1}^{K} \leftarrow$ RETRIEVETOPK(q, K), $s_1 \geq s_2 \geq \cdots \geq s_K$, $s_k = \phi(q)^\top \psi(d_k)$
$\mathcal{P} \leftarrow \{d_1, \ldots, d_{n_{\text{pos}}}\}$, $\mathcal{N} \leftarrow \{d_{K-n_{\text{neg}}+1}, \ldots, d_K\}$
// Confidence weights
$w_i^+ \leftarrow \dfrac{\exp(s_i/T)}{\sum_{i'=1}^{n_{\text{pos}}} \exp(s_{i'}/T)}$ for each $d_i \in \mathcal{P}$
$w_j^- \leftarrow \dfrac{\exp(-s_j/T)}{\sum_{j'} \exp(-s_{j'}/T)}$ for each $d_j \in \mathcal{N}$
// initialize transformation matrix
$W \leftarrow W_{\text{meta}}$
// Online gradient updates
for $t = 1$ **to** steps **do**
  $P \leftarrow \sum_{d_i \in \mathcal{P}} w_i^+ \, \phi(q)^\top W \, \psi(d_i)$
  $N \leftarrow \sum_{d_j \in \mathcal{N}} w_j^- \, \phi(q)^\top W \, \psi(d_j)$
  $m \leftarrow \alpha_{\text{mar}} + \beta_{\text{mar}} \, (1 - s_1)$
  $\mathcal{L} \leftarrow \max(0,\ m - P + N) + \lambda \, \|W - I\|_F^2$
  $W \leftarrow$ OPTIMIZERSTEP$(W,\ \nabla_W \mathcal{L},\ \eta)$
end for
$W^\star \leftarrow W$
// Update cross-query momentum
$W_{\text{ema}} \leftarrow \alpha_{\text{ema}} \, W_{\text{ema}} + (1 - \alpha_{\text{ema}}) \, W^\star$
$W_{\text{meta}} \leftarrow W_{\text{meta}} + \beta_{\text{meta}} \, (W^\star - W_{\text{meta}})$
// Rerank
return $\{d_k\}$ sorted descending by $s_{W_{\text{ema}}}(q, d_k) = \phi(q)^\top W_{\text{ema}} \, \psi(d_k)$

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate on six BEIR benchmark datasets (Thakur et al., 2021) spanning diverse domains: biomedical literature (NFCorpus, SCIDOCS, SciFact), financial QA (FiQA), argument retrieval (ArguAna), and biomedical COVID-19 retrieval (TREC-COVID). Corpus sizes range from 3.6K to 171K documents, and dense retrieval baselines (NDCG@10) vary from 0.197 (SCIDOCS) to 0.720 (SciFact), providing a challenging testbed for zero-resource generalization.

Base retriever. We use BGE-small-en-v1.5 (Xiao et al., 2024) (dimension $d = 384$, 33M parameters) as the fixed dense retriever. This small model ensures that improvements from DART are not confounded by a strong base model and matches realistic deployment constraints.

Baselines. We compare against several unsupervised or training-free methods. Dense Retrieval uses the same BGE-small encoder with cosine similarity, serving as the lower bound. PRF-Vec (Li et al., 2023) is a standard pseudo-relevance feedback method that averages top retrieved document embeddings. BM25 Rerank (Robertson and Zaragoza, 2009) reorders the dense top-100 using lexical BM25 scores, providing a purely sparse baseline. Recent training-free approaches include ASRank (Abdallah et al., 2025), ICR (Chen et al., 2024) (based on Llama-3.1-8B), and InstUPR (Huang and Chen, 2024). For reference, we also report numbers from supervised dense retrievers (e.g., ColBERT (Khattab and Zaharia, 2020), DPR-MSMARCO (Xin et al., 2022), ANCE (Xiong et al., 2020), MoDIR (Xin et al., 2022), TAS-B (Hofstätter et al., 2021), ColBERTv2 (Santhanam et al., 2022)) and the cross-encoder reranker MonoT5-

Table 1: Results on six BEIR datasets. **Abbreviations:** NFC = NFCorpus; SCI = SCIDOCS; Argu = ArguAna; COVID = TREC-COVID. * Mean per-dataset relative change versus Dense Retrieval (BGE-small); negative values for supervised methods reflect training on out-of-domain data. † Latency per query on NVIDIA RTX5090. ‡ Latency estimated from the respective papers. § ICR Avg. and Avg. Gain computed over 5 datasets (ArguAna excluded). ¶ InstUPR Avg. and Avg. Gain computed over 5 datasets (ArguAna excluded). ‖ Results for methods *not* marked with this symbol are from their original papers. **Bold** = column-wise best among all methods. "—" = not reported.

Method	NDCG@10 per Dataset						Overall		
	NFC	SCI	FiQA	Argu	COVID	SciFact	Avg.	Avg. Gain *	Latency †
Supervised Dense Retrieval									
ColBERT	0.305	0.145	0.317	0.233	0.677	0.671	0.391	−19.9%	—
DPR-MSMARCO	0.208	0.108	0.275	0.414	0.561	0.478	0.341	−31.9%	—
ANCE	0.237	0.122	0.295	0.415	0.654	0.507	0.372	−25.4%	—
MoDIR	0.244	0.124	0.296	0.418	0.676	0.502	0.377	−24.4%	—
TAS-B	0.319	0.149	0.300	0.427	0.481	0.643	0.387	−19.7%	—
RocketQAv2	0.293	0.131	0.302	0.451	0.675	0.568	0.403	−18.7%	—
SPLADEv2	0.334	0.158	0.336	0.479	0.710	0.693	0.452	−8.3%	—
ColBERTv2	0.338	0.154	0.356	0.463	0.738	0.693	0.457	−7.3%	∼80ms
Supervised Reranking									
MonoT5-base	**0.378**	0.154	0.376	0.476	**0.796**	0.675	0.476	−3.1%	∼600ms
Training-free Reranking									
Dense Retrieval (BGE-small) ‖	0.337	0.197	0.385	0.595	0.665	0.720	0.483	=0.0%	<1ms
BM25 Rerank ‖	0.302	0.156	0.220	0.371	0.685	0.588	0.387	−21.2%	<2ms
ASRank	0.346	0.184	0.352	0.478	0.737	0.710	0.468	−3.8%	∼200ms ‡
ICR	0.347	0.171	0.381	—	0.728	**0.761**	0.478 §	+0.8% §	∼200ms ‡
InstUPR	0.352	0.190	**0.398**	—	0.730	0.713	0.477 ¶	+2.6% ¶	∼200ms ‡
Test-time Adaptation									
PRF-Vec (n=3) ‖	0.347	0.203	0.371	0.602	0.663	0.710	0.483	+0.3%	<2ms
PRF-Vec (n=5) ‖	0.341	0.201	0.362	0.585	0.671	0.704	0.477	−1.0%	<1ms
DART (Ours) ‖	**0.354**	**0.205**	0.389	**0.605**	0.670	0.719	**0.490**	+2.1%	<10ms

base (Nogueira et al., 2020); these are not applicable in our zero-resource setting but illustrate the potential of supervised training.

Hyperparameters. All hyperparameters are fixed based on NFCorpus: $n_{\text{pos}} = 5$ is the number of pseudo-positive documents, $n_{\text{neg}} = 20$ is the number of pseudo-negative documents, $K = 100$ is the initial retrieval depth, $T = 0.1$ is the temperature for confidence weighting, $\alpha_{\text{mar}} = 0.1$ and $\beta_{\text{mar}} = 0.2$ are the base margin and adaptation strength, $\alpha_{\text{ema}} = 0.9$ and $\beta_{\text{meta}} = 0.1$ are the EMA decay rate and meta learning rate for cross-query momentum, $\lambda = 10^{-3}$ is the regularization coefficient, steps $= 5$ is the number of gradient updates per query, and the learning rate $\eta = 10^{-2}$. The optimizer is selected per dataset following the guidelines in Section 3.5. No dataset-specific tuning is performed.

4.2 Main Results

DART improves over the dense retrieval baseline on five of six datasets, achieving a mean per-dataset relative gain of +2.1% NDCG@10 (Table 1). The largest improvement is on NFCorpus (+5.0%), where the baseline is weakest, with further notable gains on SCIDOCS (+4.1%) and ArguAna (+1.7%). Modest gains are observed on FiQA (+1.0%) and TREC-COVID (+0.8%). SciFact is the only dataset where DART ties the baseline (−0.1%, effectively no change), likely because the high baseline score (0.720) leaves little headroom for unsupervised adaptation.

Compared to PRF-Vec, which degrades on FiQA and TREC-COVID and provides near-zero average gain (+0.3% for n=3), DART delivers consistent improvements. BM25 Rerank is unreliable, helping only on TREC-COVID while degrading by −26% on average across the remaining five datasets; DART outperforms it by +42% on those five datasets.

Recent training-free LLM-based approaches (ASRank, ICR, InstUPR) show average gains of −3.8%, +0.8%, and +2.6% respectively, but require approximately 200ms per query (∼20× the latency of DART). In contrast, DART runs in under 10ms per query on an NVIDIA RTX5090 GPU, making it far better suited for real-time, latency-

Table 2: Ablation study on four BEIR datasets. Gains are ΔNDCG@10 relative to Dense Retrieval.

Variant	**NFCorpus**		**SCIDOCS**		**FiQA**		**ArguAna**	
	NDCG@10	**Gain**	**NDCG@10**	**Gain**	**NDCG@10**	**Gain**	**NDCG@10**	**Gain**
Dense Retrieval	0.337	$=0.0\%$	0.197	$=0.0\%$	0.385	$=0.0\%$	0.595	$=0.0\%$
Base online update (conf. weighting)	0.346	+2.7%	0.199	+1.0%	0.363	–5.7%	0.595	0.0%
+ AdaMargin	0.350	+3.9%	0.201	+2.0%	0.362	–6.0%	0.595	0.0%
+ EMA	0.351	+4.0%	0.199	+1.0%	0.378	–1.8%	0.596	+0.2%
+ MetaInit	0.348	+3.3%	0.197	0.0%	0.362	–6.0%	0.599	+0.7%
+ EMA + AdaMargin	0.355	+5.3%	0.203	+3.0%	0.378	–1.8%	0.597	+0.3%
+ EMA + MetaInit	0.349	+3.6%	0.197	0.0%	0.377	–2.1%	0.599	+0.7%
+ EMA + AdaMargin + MetaInit	0.353	+4.7%	0.202	+2.5%	0.377	–2.1%	0.605	+1.7%
+ EMA + AdaMargin + MetaInit + Lion	0.354	+5.0%	0.205	+4.1%	0.389	+1.0%	0.605	+1.7%

Table 3: Statistics of $\|\Delta W\|_F$ on NFCorpus.

Statistic	Min	25%	Median	75%	Max
Value	0.000	0.048	0.095	0.111	0.125

sensitive deployment.

Notably, DART surpasses all supervised dense retrievers except ColBERTv2 and SPLADEv2 despite using no training data. Figure 2 visualises the full gain distribution across all methods: DART is the only training-free method with no negative outlier on any dataset.

4.3 Ablation Study

We evaluate the contribution of each component on a representative subset of BEIR: NFCorpus, SCIDOCS, FiQA, and ArguAna. Table 2 reports NDCG@10 for variants that incrementally add the modules described in Section 3, using steps=5. The base online update refers to confidence-weighted pseudo labels with a fixed margin. Gains are relative percentages over the dense retrieval baseline of each dataset.

EMA is the most universally beneficial component, yielding positive gains on all four datasets and single-handedly recovering FiQA from -5.7% (base) to -1.8%; this aligns with the query difficulty analysis (Section 5.3), which shows that easy queries (high s_{top1}) benefit most from smoothing rather than aggressive per-query adaptation. AdaMargin contributes most on NFCorpus, where the wide spread of query difficulty (s_{top1} range 0.5–0.9) makes fixed-margin training suboptimal. Lion provides the largest single-step lift on SCIDOCS ($+4.1\%$) and FiQA ($+1.0\%$) when added last, consistent with its advantage on clean pseudo-label distributions where sign-based updates (Chen et al., 2023) converge faster than SGD under a small step budget. The full DART model achieves the highest average gain, confirming that the three components are complementary rather than redundant.

5 Analysis

5.1 What Does W Learn?

We denote the update of the transformation matrix after processing a query as

$$\Delta W = W^* - I, \tag{17}$$

where W^* is the matrix obtained after online adaptation. The Frobenius norm $\|\Delta W\|_F$ measures the magnitude of the update.

We analyze the average update over 50 randomly sampled test queries:

$$\overline{\Delta W} = \frac{1}{n}\sum_{i=1}^{n} \Delta W_i. \tag{18}$$

Its singular value decomposition $\overline{\Delta W} = U\Sigma V^\top$ reveals a clear low-rank structure. The singular value matrix Σ (only the largest few entries shown) is

$$\Sigma = \begin{bmatrix} 0.0116 & 0 & 0 & \cdots \\ 0 & 0.0083 & 0 & \cdots \\ 0 & 0 & 0.0051 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \tag{19}$$

where $\sigma_1 = 0.0116$ accounts for 19.4% of the total variation, and the top three singular values cumulatively explain 28.4% of the variance. In contrast, a random matrix with the same Frobenius norm would exhibit a much flatter spectrum, with each of the 384 singular values approximately 0.0010. This low-rank structure indicates that DART learns *structured* semantic adjustments—rotating the scoring geometry in a small number of task-relevant directions—rather than making arbitrary perturbations.

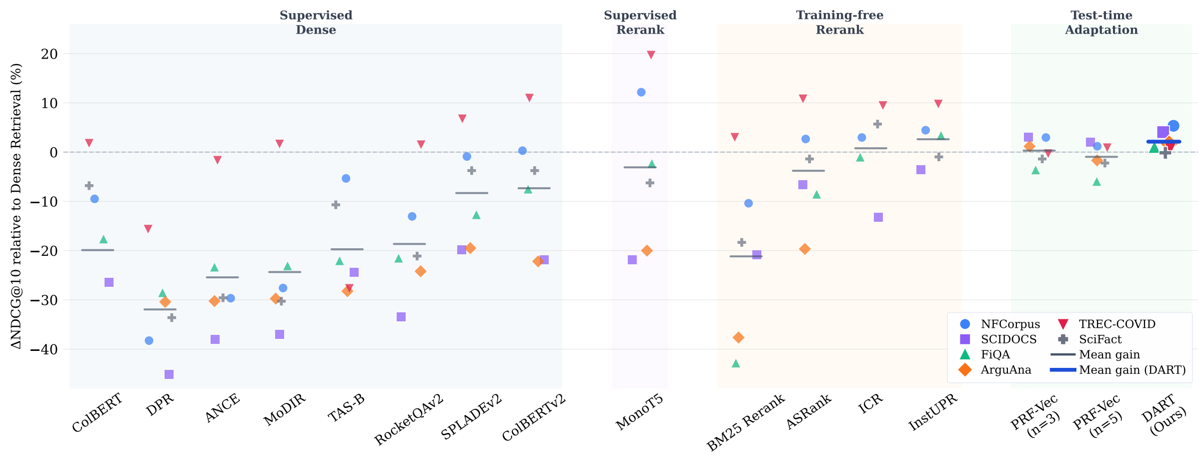

Figure 2: Per-dataset ΔNDCG@10 relative to Dense Retrieval. Each point is one dataset; bars show the mean gain.

Table 4: Average $\|\Delta W\|_F$ by s_{top1} on NFCorpus.

Interval	$[0.5, 0.6)$	$[0.6, 0.7)$	$[0.7, 0.8)$	$[0.8, 1.0)$
Mean $\|\Delta W\|_F$	0.107	0.108	0.081	0.060

5.2 How Much Does W Change?

The Frobenius norm of the identity matrix is

$$\|I\|_F = \sqrt{d} = \sqrt{384} \approx 19.6. \tag{20}$$

The updates ΔW remain very small in comparison. Table 3 summarizes the distribution of $\|\Delta W\|_F$ across queries. The median update norm is only 0.095, about 0.5% of $\|I\|_F$, confirming that the regularization term successfully constrains adaptation. The distribution is right-skewed, reflecting the heterogeneity of query difficulty.

5.3 How Does Query Affect W Change?

Grouping queries by their top-1 retrieval similarity s_{top1} reveals a basically monotonic relationship. Table 4 reports the average $\|\Delta W\|_F$ for four similarity intervals. Difficult queries (lower s_{top1}) receive larger updates, validating the adaptive margin heuristic: the model naturally allocates more adaptation capacity to queries that need it most.

6 Conclusion

We presented DART, a zero-resource reranking framework that adapts a bilinear scoring matrix at inference time using confidence-weighted pseudo-labels derived directly from the dense retrieval ranking. On six BEIR benchmarks, DART achieves a mean per-dataset relative NDCG@10 gain of +2.1% over the dense retrieval baseline (under 10ms latency per query), demonstrating a powerful capability for zero-shot performance enhancement and cross-domain generalization. The cross-query momentum mechanism (MetaInit and EMA) improves both robustness and convergence speed across the query stream. Interpretability analysis confirms that W updates are conservative, low-rank, and correlated with query difficulty. Promising future directions include session-based retrieval and cross-lingual retrieval, where the semantic-lexical gap is even more pronounced.

Limitations

The dataset-adaptive optimizer selection strategy requires processing 50–100 queries with both SGD and Lion before committing to one. In practice, SGD is the safer default: its conservative updates yield neutral-to-positive gains across all tested datasets. Lion is more aggressive—it can deliver larger improvements on clean pseudo-label distributions (e.g., NFCorpus, FiQA) but may produce zero or negative gains on noisier settings such as TREC-COVID. In truly single-pass or streaming deployments where this warm-up is undesirable, we recommend defaulting to SGD.

A second limitation concerns scalability to larger encoders. DART currently optimizes a full $d \times d$ matrix W, whose memory and per-query computation grow quadratically with the embedding dimension. For encoders with $d \geq 768$ (e.g., large BERT-family models), a low-rank parameterization $W = I + AB^\top$ with $A, B \in \mathbb{R}^{d \times r}$ and $r \ll d$ would substantially reduce overhead without sacrificing performance; we leave this extension to future work.

Acknowledgments

We would like to thank the anonymous reviewers for their insightful comments and constructive suggestions, which helped improve the quality of this paper.

References

Abdelrahman Abdallah, Jamshid Mozafari, Bhawna Piryani, and Adam Jatowt. 2025. Asrank: Zero-shot re-ranking with answer scent for document retrieval. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 2950–2970.

Ekin Akyürek, Mehul Damani, Adam Zweiger, Linlu Qiu, Han Guo, Jyothish Pari, Yoon Kim, and Jacob Andreas. 2024. The surprising effectiveness of test-time training for few-shot learning. *arXiv preprint arXiv:2411.07279*.

Shijie Chen, Bernal Jiménez Gutiérrez, and Yu Su. 2024. Attention in large language models yields efficient zero-shot re-rankers. *arXiv preprint arXiv:2410.02642*.

Xiangning Chen, Chen Liang, Da Huang, Esteban Real, Kaiyuan Wang, Hieu Pham, Xuanyi Dong, Thang Luong, Cho-Jui Hsieh, Yifeng Lu, and 1 others. 2023. Symbolic discovery of optimization algorithms. *Advances in neural information processing systems*, 36:49205–49233.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Jun-Mei Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiaoling Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 179 others. 2025. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645:633 – 638.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

Jiafeng Guo, Yixing Fan, Liang Pang, Liu Yang, Qingyao Ai, Hamed Zamani, Chen Wu, W Bruce Croft, and Xueqi Cheng. 2020. A deep look into neural ranking models for information retrieval. *Information Processing & Management*, 57(6):102067.

Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. 2021. Efficiently teaching an effective dense retriever with balanced topic aware sampling. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 113–122, New York, NY, USA. Association for Computing Machinery.

Chao-Wei Huang and Yun-Nung Chen. 2024. Instupr: Instruction-based unsupervised passage reranking with large language models. *arXiv preprint arXiv:2403.16435*.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *Trans. Mach. Learn. Res.*, 2022.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE transactions on big data*, 7(3):535–547.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pages 6769–6781.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48.

Victor Lavrenko and W Bruce Croft. 2017. Relevance-based language models. In *ACM SIGIR Forum*, volume 51, pages 260–267. ACM New York, NY, USA.

Hang Li, Ahmed Mourad, Shengyao Zhuang, Bevan Koopman, and Guido Zuccon. 2023. Pseudo relevance feedback with deep language models and dense retrievers: Successes and pitfalls. *ACM Transactions on Information Systems*, 41(3):1–40.

Hang Li, Xiao Wang, Bevan Koopman, and Guido Zuccon. 2025. Pseudo relevance feedback is enough to close the gap between small and large dense retrieval models. *arXiv preprint arXiv:2503.14887*.

Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. 2022. *Pretrained transformers for text ranking: Bert and beyond*. Springer Nature.

Yuejiang Liu, Parth Kothari, Bastien Van Delft, Baptiste Bellot-Gurlet, Taylor Mordan, and Alexandre Alahi. 2021. Ttt++: When does self-supervised test-time training fail or thrive? *Advances in Neural Information Processing Systems*, 34:21808–21820.

Rui Meng, Ye Liu, Semih Yavuz, Divyansh Agarwal, Lifu Tu, Ning Yu, Jianguo Zhang, Meghana Bhat, and Yingbo Zhou. 2022. Unsupervised dense retrieval deserves better positive pairs: Scalable augmentation with query extraction and generation. *arXiv preprint arXiv:2212.08841*.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*.

Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. 2020. Document ranking with a pretrained sequence-to-sequence model. In *Findings of the association for computational linguistics: EMNLP 2020*, pages 708–718.

Ronak Pradeep, Sahel Sharifymoghaddam, and Jimmy Lin. 2023. Rankvicuna: Zero-shot listwise document reranking with open-source large language models. *arXiv preprint arXiv:2309.15088*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 3982–3992.

Stephen Robertson and Hugo Zaragoza. 2009. *The probabilistic relevance framework: BM25 and beyond*, volume 4. Now Publishers Inc.

Jon Saad-Falcon, Omar Khattab, Keshav Santhanam, Radu Florian, Martin Franz, Salim Roukos, Avirup Sil, Md Sultan, and Christopher Potts. 2023. Udapdr: unsupervised domain adaptation via llm prompting and distillation of rerankers. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 11265–11279.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022. Colbertv2: Effective and efficient retrieval via lightweight late interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3715–3734.

Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. 2023. Is chatgpt good at search? investigating large language models as re-ranking agents. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 14918–14937.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. 2020. Test-time training with self-supervision for generalization under distribution shifts. In *International conference on machine learning*, pages 9229–9248. PMLR.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663*.

Kexin Wang, Nandan Thakur, Nils Reimers, and Iryna Gurevych. 2022a. Gpl: Generative pseudo labeling for unsupervised domain adaptation of dense retrieval. In *Proceedings of the 2022 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 2345–2360.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022b. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.

Xiao Wang, Craig Macdonald, Nicola Tonellotto, and Iadh Ounis. 2023. Colbert-prf: Semantic pseudo-relevance feedback for dense passage and document retrieval. *ACM Transactions on the Web*, 17(1):1–39.

Orion Weller, Kathryn Ricci, Eugene Yang, Andrew Yates, Dawn Lawrie, and Benjamin Van Durme. 2025. Rank1: Test-time compute for reranking in information retrieval. *arXiv preprint arXiv:2502.18418*.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. 2024. C-pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*, pages 641–649.

Ji Xin, Chenyan Xiong, Ashwin Srinivasan, Ankita Sharma, Damien Jose, and Paul Bennett. 2022. Zero-shot dense retrieval with momentum adversarial domain invariant representations. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 4008–4020.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808*.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2023. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*.

Shengyao Zhuang, Linjun Shou, and Guido Zuccon. 2023. Augmenting passage representations with query generation for enhanced cross-lingual dense retrieval. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1827–1832.