Q-REACT: QUERY RESIDUAL EMBEDDING ADAPTATION THROUGH CONDITIONAL TRANSFORMATIONS FOR VISUAL DOCUMENT RETRIEVAL

Anonymous authors
Paper under double-blind review

ABSTRACT

Visual document retrieval (VDR) relies on an offline constructed page index, yet improving retrieval against this fixed index often requires costly model training or access to encoder parameters, which may be unavailable for proprietary APIs. Multimodal rerankers provide high-quality relevance signals, but under limited budgets they assess only a small subset of queries and candidate pages. Their gains therefore remain local to candidate reordering, leaving useful supervision unused for improving the retrieval system as a whole. We propose Q-REACT (Query Residual Embedding Adaptation through Conditional Transformations), a test-time optimization method that reuses reranker rewards to improve retrieval over the full index with limited feedback. Q-REACT combines query-dependent low-rank residual adaptation with document-context scoring while keeping the encoders and page index fixed. It distills local reranker preferences using cross-entropy with a student distribution normalized over the complete task-specific index, allowing unscored pages to participate as competitors through cached embeddings without additional reward calls. Experiments across eight ViDoRe V3 tasks and five open-weight and proprietary backbones show higher average retrieval quality than direct reranking and TTT-Embed at representative sparse and full-coverage feedback budgets. Our findings show that Q-REACT makes more effective use of limited reranker feedback, improving retrieval across the query collection without retraining encoders or rebuilding the page index.

1 INTRODUCTION

Visual document retrieval (VDR) encodes and indexes complete pages offline, preserving text, tables, images, and layout without requiring OCR or manual parsing (Wang et al., 2025; Tanaka et al., 2025; Masry et al., 2025). A standard approach to improving query–page alignment is to retrain or fine-tune the multimodal embedding model. However, this requires access to model parameters and training resources and may entail re-encoding the entire page corpus. This route is unavailable for proprietary embedding APIs that expose no parameter access. Multimodal rerankers offer an alternative by providing finer relevance judgments over retrieved candidates (Hu et al., 2026; Lin et al., 2025). Yet conventional reranking produces a better ordering of the current query's candidates but leaves the resulting relevance feedback unused beyond that ranking. Under limited feedback budgets, its benefits are therefore restricted to the queries selected for reranking. As Figure 1 shows, sparse query coverage yields negligible average improvements in overall retrieval performance. This raises the question of whether finite reranker feedback can be used to improve overall retrieval performance without updating either the encoders or the page index.

Residual learning (He et al., 2016) and query-side test-time optimization (Sung et al., 2023; Reddy et al., 2023) offer a route to adapting retrieval without modifying the encoder or re-encoding the corpus. TTT-Embed (Chen & Wu, 2026) establishes this approach through regularized distillation of reranker feedback into reusable residual vectors. Its Global and Task modes share one fixed vector within each scope and fit preferences over rewarded candidates. However, information-rich document pages in VDR can be relevant to diverse queries seeking different textual and visual evidence (Faysse et al., 2025; Cai et al., 2025). A single shared residual may therefore struggle to accommodate these heterogeneous query–page matching requirements. In our ViDoRe V3 eval-

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.

uation (Loison et al., 2026), Figure 1 shows that TTT-Embed can degrade retrieval under sparse feedback and yields limited gains at full coverage, consistent with the constrained capacity of a shared residual. This motivates VDR adaptation that delivers stable gains with sparse feedback and larger improvements with broader supervision.

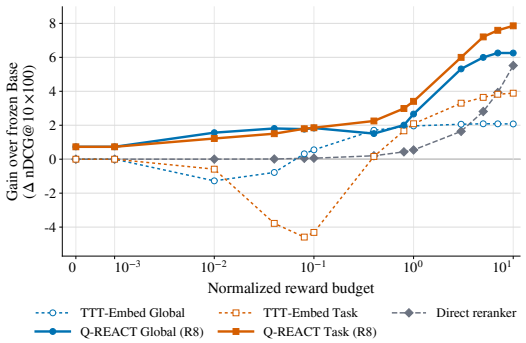


Figure 1: Reward-budget scaling averaged across five frozen VDR backbones.

To address these challenges, we propose Q-REACT, which reuses reranker rewards to learn a shared low-rank query transformation on top of a document-structure prior. Unlike the fixed residual vectors shared by TTT-Embed's Global and Task modes, Q-REACT produces query-dependent residuals through a low-rank transformation, accommodating diverse query intents in VDR. Moreover, feedback covers only a small candidate set, whereas adapted queries retrieve from the complete page index, where pages may share similar content or layouts. Building on TTT-Embed's regularized distillation formulation, we normalize the student distribution over the full task-specific index, bringing unscored pages into competition through cached embeddings without additional reward calls. This exposes query updates to competing pages beyond the feedback set. To provide a stronger ranking foundation under sparse feedback, motivated by the degradation observed for TTT-Embed in Figure 1, we further incorporate document structure into retrieval scoring. We add the strongest query match within each source document to every page's direct relevance. Applied to the original query embeddings, this rule provides a feedback-free retrieval prior on which residual adaptation builds. Our contributions are threefold. **(1) Query-dependent low-rank adaptation.** We introduce a shared low-rank query transformation that accommodates diverse VDR queries and supports larger retrieval gains as more reranker feedback becomes available, while keeping encoders and the page index frozen. **(2) Document-aware adaptation from reusable feedback.** We integrate a feedback-free document-structure prior with reranker-guided adaptation, providing a stable ranking foundation under sparse feedback. We reuse local reranker rewards through regularized distillation, accounting for full-index competition during alignment. **(3) Retrieval performance and empirical insights.** Across eight ViDoRe V3 tasks and five backbones, Q-REACT outperforms the evaluated baselines on average at representative sparse and full-coverage budgets and generalizes better than TTT-Embed to held-out queries and tasks. Ablations favor query-dependent low-rank transformations over fixed residuals and show that full-index normalization better exploits task-specific low-rank capacity at full feedback coverage.

2 Related Work

Visual document retrieval and multimodal reranking. Visual document retrieval independently encodes queries and complete pages, preserving text, tables, images, and layout while supporting offline page indexing (Faysse et al., 2025; Macé et al., 2025; Li et al., 2026b). Recent systems include single-vector retrievers such as Qwen3-VL-Embedding and WeMM-Embedding (Li et al., 2026a; Zhou et al., 2026), as well as multi-vector late-interaction models that enable finer-grained matching at higher storage and computation costs (Faysse et al., 2025; Moreira et al., 2026; Liu et al., 2026). Multimodal rerankers further refine retrieved candidates through cross-encoder, prompt-based, or listwise scoring (Li et al., 2026a; Saha & Gokhale, 2026; Chen et al., 2025; Sun et al., 2026). These approaches provide test-time relevance judgments that improve candidate ranking without modifying the initial retriever. Q-REACT studies how a finite budget of such judgments can be incorporated into a document-aware retrieval function, combining structural evidence with query adaptation while preserving the encoders and cached page embeddings.

Test-time and query-side optimization. Test-time training and scaling improve predictions through adaptation or additional inference computation (Sun et al., 2020; Liu et al., 2021; Snell et al., 2024; Maekawa et al., 2026). In retrieval, ANCE-PRF trains a query encoder to incorporate information from top-ranked documents while retaining the document index (Yu et al., 2021).

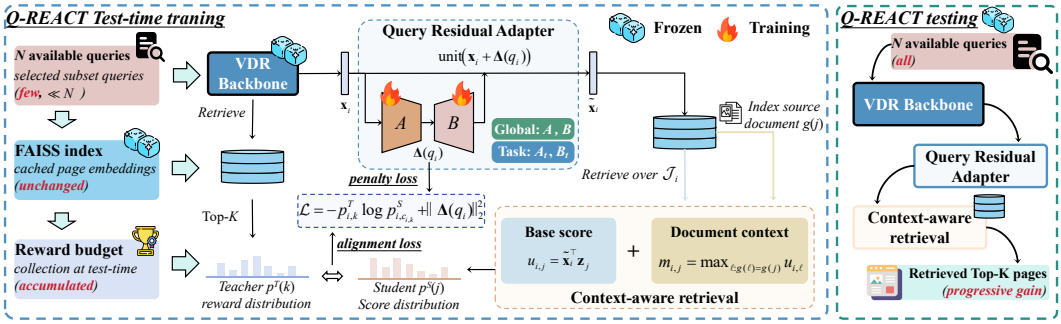


Figure 2: Overview of Q-REACT. Local reranker preferences supervise a low-rank query adapter through full-index distillation and a residual penalty. The adapted query determines both direct page relevance and document-level evidence, using the same scoring rule during adaptation and inference.

Reranker-guided methods such as TOUR and ReFIT instead refine individual query representations during inference (Sung et al., 2023; Reddy et al., 2023; Gangi Reddy et al., 2025). GQR uses scores from a complementary retriever to guide query optimization, enabling hybrid retrieval through representation refinement (Uzan et al., 2026). However, most methods that optimize queries at test time do so independently for each query, making broad query coverage difficult under a limited feedback budget and increasing inference costs through repeated optimization. Most closely related, TTT-Embed distills reranker rewards into additive residuals under Global, Task, and Query sharing scopes (Chen & Wu, 2026). Its Global and Task configurations apply a shared fixed residual within each scope. In contrast, Q-REACT learns a low-rank transformation that produces query-dependent corrections to accommodate the more complex VDR setting.

3 Method

Q-REACT learns query-dependent low-rank residuals through full-index distillation of reranker feedback with document-context scoring, keeping encoders and page embeddings frozen. For each backbone, **Global** pools feedback to learn one transformation across tasks, while **Task** learns one per task. Both use the same residual form, scoring rule, and objective.

3.1 Problem Formulation

Given query and page collections $\mathcal{Q} = \{q_i\}_{i=1}^{N}$ and $\mathcal{P} = \{p_j\}_{j=1}^{P}$, we aim to improve retrieval using a finite budget of reranker feedback. Frozen encoders f_Q and f_D map queries and pages to $\mathbb{R}^D$. With $\text{unit}(\mathbf{a}) = \mathbf{a}/\|\mathbf{a}\|_2$ for nonzero $\mathbf{a}$, let $\mathbf{x}_i = \text{unit}(f_Q(q_i))$ and $\mathbf{z}_j = \text{unit}(f_D(p_j))$. Page vectors are computed once and cached. For a query from task t, $\mathcal{J}_i = \mathcal{J}_t \subseteq \{1, \ldots, P\}$ denotes the complete searchable page index. Both sharing modes use this same task-specific retrieval scope. Base retrieval ranks pages by $s_{i,j}^{(0)} = \mathbf{x}_i^\top \mathbf{z}_j$. Query adaptation introduces a residual,

$$\tilde{\mathbf{x}}_i = \text{unit}\left(\mathbf{x}_i + \boldsymbol{\Delta}_\theta(q_i)\right), \tag{1}$$

where θ is shared within the selected scope, while $\boldsymbol{\Delta}_\theta(q_i)$ depends on the query embedding. Let $\mathcal{I} \subseteq \{1, \ldots, N\}$ index queries selected for feedback. For each $i \in \mathcal{I}$, Base returns ordered top-K page indices $\mathcal{C}_i = (c_{i,1}, \ldots, c_{i,K})$. The reranker assigns rewards $\mathcal{R}_i = (\rho_{i,1}, \ldots, \rho_{i,K})$, where $\rho_{i,k} \in \mathbb{R}$ scores page $p_{c_{i,k}}$, with higher scores indicating stronger preference. Queries, candidates, and rewards remain fixed during adaptation. The total and average per-query budgets are

$$\mathcal{B} = \sum_{i \in \mathcal{I}} |\mathcal{R}_i|, \qquad b = \frac{\mathcal{B}}{N}. \tag{2}$$

With K rewards per selected query, $\mathcal{B} = K|\mathcal{I}|$ and $b = K|\mathcal{I}|/N$. We increase query coverage at fixed K, then increase K after full coverage. This budget counts scored query–page pairs, with adaptation and retrieval computation assessed separately. Budget-scaling evaluation uses all of $\mathcal{Q}$, including rewarded queries. Reusing feedback during optimization incurs no additional reward cost,

and adaptation uses no ground-truth relevance labels. Separate held-out-query and task evaluations assess transfer without evaluation-query feedback.

3.2 Query-Dependent Low-Rank Residuals

TTT-Embed's Global and Task modes apply a common additive residual within each sharing scope (Chen & Wu, 2026). Q-REACT shares a linear map that produces a residual from each query embedding. Let $\mathbf{A} \in \mathbb{R}^{r \times D}$ and $\mathbf{B} \in \mathbb{R}^{D \times r}$, with $r \ll D$. For the selected scope, $\theta = (\mathbf{A}, \mathbf{B})$ and

$$\boldsymbol{\Delta}_\theta(q_i) = \mathbf{W}\mathbf{x}_i = \frac{1}{\sqrt{r}}\mathbf{B}\mathbf{A}\mathbf{x}_i, \qquad \operatorname{rank}(\mathbf{W}) \le r. \tag{3}$$

Writing $\mathbf{b}_h$ for column h of $\mathbf{B}$ gives $\boldsymbol{\Delta}_\theta(q_i) = r^{-1/2} \sum_{h=1}^{r} [\mathbf{A}\mathbf{x}_i]_h \mathbf{b}_h$. The columns provide shared correction directions, while $\mathbf{A}\mathbf{x}_i$ supplies query-dependent coefficients. Residual directions and magnitudes can therefore vary across queries within a common subspace of dimension at most r. The rank bound applies to the residual map, and the original query remains present in Equation 1. Each matrix pair contains $2Dr$ trainable parameters, allowing rank to control capacity relative to an unrestricted $D \times D$ residual map. The factor $1/\sqrt{r}$ follows rsLoRA's rank scaling (Kalajdzievski, 2023), applied here to an external query transformation. We initialize $\mathbf{B}$ to zero, giving $\boldsymbol{\Delta}_\theta(q_i) = \mathbf{0}$. With document-context scoring active, this yields the Context-only starting point defined below. We omit the subscript θ on residuals in the remaining equations for brevity.

3.3 Document-Context-Aware Retrieval Scoring

We use document membership as a structural prior, assuming that a strong query match within a document can provide relevance evidence for its other pages. Let $g(j)$ identify the source document of page p_j, with this metadata available for all indexed pages. For $j \in \mathcal{J}_i$, direct page relevance is $u_{i,j} = \tilde{\mathbf{x}}_i^\top \mathbf{z}_j$, and document-level evidence is $m_{i,j} = \max_{\ell \in \mathcal{J}_i,\, g(\ell) = g(j)} u_{i,\ell}$. The final score combines these terms as $s_{i,j} = u_{i,j} + m_{i,j}$. The maximum includes every indexed page in the same document, including p_j itself. Pages from one document receive the same context term, preserving their ordering under direct relevance while changing comparisons across documents. Both terms depend on the adapted query, so the shared map is trained through the same document-aware score used at inference. The aggregation uses cached embeddings and fixed document membership without additional encoding or reranker calls. Setting $\boldsymbol{\Delta}(q_i) = \mathbf{0}$ defines **Context-only**, which applies this scoring rule to the original query embeddings without feedback. By operating on base retrieval scores, this prior provides a stronger ranking foundation when limited feedback constrains adaptation.

3.4 Full-Index Distillation from Local Rewards

Following TTT-Embed's regularized distillation formulation (Chen & Wu, 2026), we train the low-rank adapter using reranker preferences. Q-REACT couples this local supervision with competition over the complete task-specific page index. This design exposes query-dependent corrections to the retrieval space in which they will be deployed, while retaining an explicit constraint on query displacement.

Teacher distribution. For each rewarded query q_i with $i \in \mathcal{I}$, we convert the reranker scores on the fixed Base top-K candidates into a distribution of relative preferences,

$$p^T_{i,k} = \frac{\exp(\rho_{i,k}/\tau_T)}{\sum_{h=1}^{K} \exp(\rho_{i,h}/\tau_T)}, \qquad i \in \mathcal{I}, \quad k = 1, \dots, K, \tag{4}$$

where $\tau_T > 0$ controls the concentration of the teacher distribution and $p^T_{i,k}$ refers to page $p_{c_{i,k}}$. The candidates and teacher probabilities remain fixed throughout adaptation at each reward budget.

Student distribution. The adapted query induces a retrieval distribution over the complete task-specific index $\mathcal{J}_i$ through the document-context-aware scores defined in Section 3.3,

$$p^S_{i,j} = \frac{\exp(s_{i,j}/\tau_S)}{\sum_{\ell \in \mathcal{J}_i} \exp(s_{i,\ell}/\tau_S)}, \qquad i \in \mathcal{I}, \quad j \in \mathcal{J}_i, \tag{5}$$

where $\tau_S > 0$ is the student temperature. The index $c_{i,k}$ links each teacher probability to its corresponding page in the student distribution. Local rewards supervise only a small candidate set, whereas the adapted query must rank the full page index. Normalizing over $\mathcal{J}_i$ brings unscored pages into competition through their cached embeddings, aligning the scope of adaptation with retrieval without additional reward calls.

Regularized query adaptation. We fit the local reward preferences using cross-entropy with the full-index student distribution. For T tasks, let $\mathcal{I}_t$ partition the rewarded query indices by task, with $1 \leq |\mathcal{I}_t| \leq N_t$, where N_t is the total query count of task t. For each $i \in \mathcal{I}_t$, the distillation loss is

$$\mathcal{L}_{\mathrm{CE},t,i} = -\sum_{k=1}^{K} p^T_{i,k} \log p^S_{i,c_{i,k}}. \tag{6}$$

Appendix D decomposes this loss into candidate-relative preference fitting and a term that increases the candidates' total probability mass within the full index. The additional term arises from the normalization scope, as CE and forward KL are equivalent for fixed teachers and matched student distributions. To control the magnitude of the learned corrections, we penalize the realized residual on each rewarded query before normalization,

$$\mathcal{L}_{\mathrm{res},t,i} = \|\mathbf{\Delta}(q_i)\|_2^2. \tag{7}$$

With a unit regularization coefficient, the complete objective is

$$\mathcal{L} = \frac{1}{T}\sum_{t=1}^{T} \frac{1}{|\mathcal{I}_t|} \sum_{i \in \mathcal{I}_t} \left(\mathcal{L}_{\mathrm{CE},t,i} + \mathcal{L}_{\mathrm{res},t,i}\right). \tag{8}$$

Averaging within tasks and then across tasks gives each task equal weight. Global optimizes a shared matrix pair using all task losses, while Task optimizes one pair per task. In this formulation, the low-rank map determines the capacity for query-dependent corrections, full-index normalization introduces competition beyond the rewarded candidates, and residual regularization controls query displacement. The low-rank ablations in Table 3 favor full-index normalization for Task but candidate normalization for Global, suggesting that task specialization helps accommodate broader competition by reducing conflicting demands across tasks.

3.5 Inference

After adaptation, the transformation matrices are fixed. For a query from task t, Global uses the shared pair $(\mathbf{A}, \mathbf{B})$, while Task uses $(\mathbf{A}_t, \mathbf{B}_t)$. We form the adapted query using Equations 1 and 3, recompute direct relevance and document-level evidence over the complete task-specific index, and return the top-K_{ret} pages under the scoring rule in Section 3.3. The inference depth K_{ret} is distinct from the feedback depth K. This new search can retrieve pages outside the original rewarded candidates. It requires no per-query optimization or additional reranker calls, and preserves both encoders and the cached page embeddings.

4 Experiments

4.1 Setup

Dataset and Evaluation Metric. We evaluate Q-REACT on ViDoRe V3 (Loison et al., 2026; Macé et al., 2025) with 14,514 queries and 19,256 pages across eight tasks. We report nDCG@10 ($\times 100$), averaging within each task and then equally across tasks for each backbone. Aggregate results average across five backbones. All budgets evaluate the same full query collection, with budget denominator $N = 14{,}514$. Feedback coverage expands through nested query subsets. For each backbone and budget, supervised methods share the same candidates and rewards, which remain fixed during optimization. Reusing feedback incurs no additional reward cost. With $K = 10$, $0 < b < 10$ gives partial query coverage, while $b = 10$ provides ten page-level rewards per query at full coverage. Budget curves track retrieval quality as coverage expands. Held-out-query and leave-one-task-out experiments separately assess generalization.

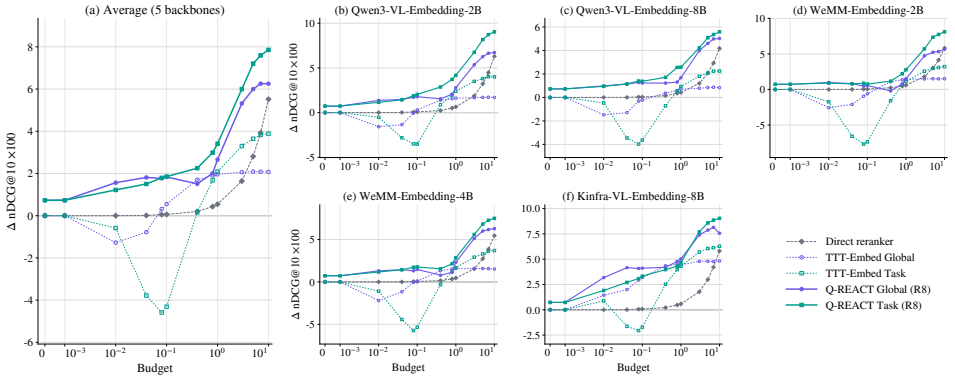


Figure 3: Reward-budget scaling across five frozen VDR backbones. Panel (a) shows the five-model average, and panels (b)–(f) show individual backbones. Solid and dotted curves compare Q-REACT and TTT-Embed in Global and Task modes, with dashed curves showing direct reranking. Q-REACT points at 0 and 0.001 use no feedback budget and show Context-only performance, which remains constant across budgets.

Table 1: nDCG@10 ($\times 100$) on ViDoRe V3 at two feedback budgets. Reranker, TTT-Embed, and Q-REACT share the same feedback within each budget. Base and GQR are independent of this budget, so their results are repeated. GQR (ICLR 2026) refines every query for 50 steps with text-embeddings feedback. Bold marks the best score in each row.

Backbone	Base	Reranker	TTT-Embed		GQR	Q-REACT	
			Global	Task		Global	Task
$b = 0.1$ (145 rewarded queries, approximately 1% coverage)							
Qwen3-VL-Embedding-2B	52.80	52.88	53.10	49.28	54.20	54.61	**54.76**
Qwen3-VL-Embedding-8B	58.99	59.03	58.76	55.36	54.89	60.34	**60.46**
WeMM-Embedding-2B	53.03	53.08	52.42	45.66	**54.15**	53.68	53.85
WeMM-Embedding-4B	55.41	55.46	55.50	50.07	54.89	56.86	**57.28**
Kinfra-VL-Embedding-8B	53.42	53.50	56.62	51.71	53.63	**57.51**	56.71
Average	54.73	54.79	55.28	50.42	54.35	56.60	**56.61**
$b = 10$ (14,514 rewarded queries, full coverage)							
Qwen3-VL-Embedding-2B	52.80	59.10	54.49	56.79	54.20	59.68	**61.80**
Qwen3-VL-Embedding-8B	58.99	63.18	59.83	61.24	54.89	64.07	**64.62**
WeMM-Embedding-2B	53.03	58.85	54.53	56.24	54.15	58.78	**61.17**
WeMM-Embedding-4B	55.41	60.87	56.93	59.11	54.89	61.64	**62.88**
Kinfra-VL-Embedding-8B	53.42	59.23	58.24	59.70	53.63	61.53	**62.57**
Average	54.73	60.25	56.81	58.62	54.35	61.14	**62.61**

Baselines. We use five frozen single-vector VDR backbones. The four open-weight models are Qwen3-VL-Embedding-2B, Qwen3-VL-Embedding-8B (Li et al., 2026a), WeMM-Embedding-2B, and WeMM-Embedding-4B (Zhou et al., 2026). The proprietary backbone is Kinfra-VL-Embedding-8B from Tencent Cloud[1]. For each backbone, **Base** uses the original frozen embeddings, **Reranker** applies the multimodal Qwen3-VL-Reranker-8B (Li et al., 2026a), and **TTT-Embed** (Chen & Wu, 2026) provides the test-time query optimization baseline. We also include **GQR** (Uzan et al., 2026), which independently optimizes each query representation using guidance from a text embedding model, without reranker feedback. As part of the supplementary analysis, we additionally conducted experiments with Qwen3-VL-Reranker-2B, Qwen3-VL-8B, and Gemma-4-12B-it (Team et al., 2026) as rerankers. Implementation details are provided in Appendix A.

[1] https://intl.cloud.tencent.com/zh/products/tokenhub

4.2 Retrieval Performance Across Feedback Budgets

Retrieval gains under limited feedback. Figure 3 shows positive gains for Q-REACT across $b \in [0.01, 0.1]$, approaching 2 nDCG@10 points in both Global and Task modes near the upper end, while direct reranking yields negligible average gains. As an integral component of Q-REACT, the document-context scoring module provides a strong structural foundation on which query adaptation builds, supporting consistent gains even under sparse feedback. With feedback from only 145 of 14,514 queries, Q-REACT achieves these gains using less than 1% of the budget required for full query coverage. At $b = 1$, its gains reach roughly 3 points, compared with about half a point for direct reranking. Through query adaptation, Q-REACT incorporates reranker signals that would otherwise be discarded after candidate reordering into the retrieval function, improving overall ranking quality beyond direct reranking under the same feedback budget.

Budget utilization through low-rank adaptation. Figure 3(a) shows that Q-REACT translates increasing feedback into larger retrieval gains. At the sparsest budgets below $b = 1$, Global initially leads by concentrating limited supervision in one low-rank module, while Task distributes feedback across multiple modules. As coverage expands, each Task module receives more supervision for task-specific corrections, and Task overtakes Global. With the adapter rank held fixed, this pattern is consistent with richer feedback enabling progressively fuller use of the available transformation capacity to accommodate heterogeneous VDR queries. At full query coverage ($b = 10$), Table 1 reports Q-REACT gains of 6.41 and 7.88 points, exceeding TTT-Embed's 2.08 and 3.89 and direct reranking's 5.52. These results support the ability of Q-REACT's query-dependent low-rank adaptation to convert additional feedback into more effective multimodal matching, highlighting its budget utilization and suitability for diverse VDR tasks.

4.3 Comparison with Per-Query Optimization

Retrieval quality under limited feedback. Table 1 compares Q-REACT with direct reranking, TTT-Embed, and GQR (Uzan et al., 2026) on the full collection of 14,514 queries, including rewarded queries. At $b = 0.1$, feedback-based methods receive ten reranker judgments for each of 145 queries, while GQR independently optimizes every query using text-embedding guidance. Q-REACT achieves average scores of 56.60 and 56.61 in Global and Task modes, improving over Base by 1.87 and 1.88 points and exceeding GQR by 2.25 and 2.26 points, respectively. These results demonstrate higher overall retrieval quality under a limited feedback budget. Global uses a single low-rank residual module and requires no per-query optimization at inference, reducing the computational overhead associated with GQR's per-query updates.

Performance at full feedback coverage. At $b = 10$, every query receives ten reranker judgments. Q-REACT achieves average scores of 61.14 and 62.61 in Global and Task modes, exceeding direct reranking by 0.90 and 2.36 points and the corresponding TTT-Embed variants by 4.34 and 3.99 points, respectively. GQR averages 54.35 despite optimizing every query for 50 steps, and both Q-REACT modes outperform it on all five backbones. These results show that shared query adaptation can achieve higher retrieval quality than independent per-query optimization, while incorporating reranker rewards more effectively than direct candidate reordering.

Table 2: Generalization without evaluation-query feedback. Bold marks the best score within each setting. Context-only and TTT-Embed rows report five-backbone averages. Context-only has no G/T modes.

Backbone	Held-out queries			LOTO	
	Base	G	T	Base	G
Qwen3-VL-Embedding-2B	54.51	**57.15**	56.21	52.80	**53.96**
Qwen3-VL-Embedding-8B	60.48	**61.53**	60.98	58.99	**59.11**
WeMM-Embedding-2B	54.32	**56.59**	54.30	53.03	**54.13**
WeMM-Embedding-4B	56.42	**58.99**	57.92	55.41	**56.80**
Kinfra-VL-Embedding-8B	54.96	**60.30**	58.30	53.42	**57.52**
Average	56.14	**58.91**	57.54	54.73	**56.31**
TTT-Embed (average)	56.14	58.03	57.00	54.73	55.70
Context-only (average)	56.14	57.11		54.73	55.46

4.4 Generalization to Held-Out Queries and Tasks

We train on 60% of each task's original queries with ten page-level rewards per query ($b_{\text{train}} = 10$) and evaluate on the remaining 40%. In leave-one-task-out (LOTO) evaluation, Global trains on seven tasks' training pools and evaluates on the remaining task, averaging over eight folds. Evalu-

ation queries provide no feedback, and encoders, page indices, and learned adapters remain frozen. In Table 2, Context-only reaches 57.11 on held-out queries and 55.46 under LOTO, gaining 0.97 and 0.73 points over Base. Full Q-REACT reaches 58.91 and 57.54 on held-out queries in Global and Task modes, adding 1.80 and 0.43 points over Context-only. LOTO Global reaches 56.31, a further gain of 0.85 points. Q-REACT exceeds TTT-Embed in all three comparisons. Global improves all five backbones over Base in both settings, while Task improves four on held-out queries. Global pools all feedback in one residual module, whereas Task distributes supervision across separate modules, helping explain Global's stronger generalization under these protocols. These results support generalization of Q-REACT's query-dependent low-rank transformations to held-out queries and tasks under the specified training-pool and feedback settings.

5 Analysis

5.1 Ablation Studies

We examine regularization and document-context scoring across feedback budgets, then jointly vary residual parameterization, normalization scope, and document context at full feedback coverage.

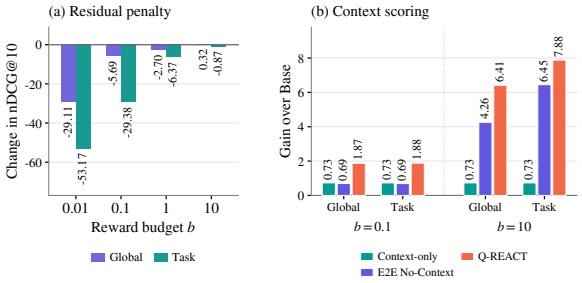


Figure 4: Ablations averaged across five backbones. (a) Score changes after removing residual regularization. (b) Gains over Base for Context-only, adaptation without context, and full Q-REACT.

Regularization under Sparse Feedback. Figure 4(a) shows that residual regularization is essential for stable low-rank adaptation under sparse supervision. Removing the penalty reduces Global and Task scores by 29.11 and 53.17 points at $b = 0.01$, and by 5.69 and 29.38 points at $b = 0.1$. The penalty constrains query-dependent corrections when limited feedback cannot adequately prevent excessive displacement from frozen representations. At $b = 10$, removing regularization changes performance by only $+0.32$ and -0.87 points, consistent with richer supervision providing stronger relevance constraints and reducing adaptation drift. Regularization matters most when feedback is sparse.

Context as a Structural Foundation. Figure 4(b) demonstrates the importance of Q-REACT's internal context-scoring module under limited feedback. At $b = 0.1$, adaptation without context gains only 0.69 points in either mode, while Context-only provides a feedback-free gain of 0.73 points. Full Q-REACT reaches 1.87 and 1.88 points, showing that query adaptation benefits from operating within a retrieval function grounded in document structure. Context thus provides a stable ranking foundation and complements the corrections learned from sparse rewards. At $b = 10$, richer supervision allows adaptation without context to achieve gains of 4.26 and 6.45 points. Retaining context raises these gains to 6.41 and 7.88 points, respectively. These $b = 10$ results correspond to the low-rank Full entries in Table 3. The structural foundation remains beneficial as feedback increases, even when learned corrections already provide substantial improvements independently.

Table 3: Controlled factorial ablation at $b = 10$, averaged across five backbones (nDCG@10 $\times 100$). Top-10 and Full normalize student scores over rewarded candidates and the full task index, respectively. Bold marks the best score in each numeric column.

Normalization	Context off		Context on	
	Global	Task	Global	Task
Fixed vector				
Top-10	56.30	58.09	57.64	59.76
Full	56.23	57.40	57.58	59.14
Low-rank ($r = 8$)				
Top-10	**59.74**	59.24	**61.52**	61.47
Full	58.99	**61.18**	61.14	**62.61**

Low-Rank Capacity and Normalization Scope. Table 3 isolates the effects of residual parameterization and normalization through comparisons with matched context settings at $b = 10$. Low-rank residuals outperform fixed vectors in all eight comparisons of five-backbone averages, with gains of

1.14–3.89 points across both normalization scopes, context settings, and sharing modes. This consistent advantage supports the effectiveness of query-dependent low-rank adaptation in the evaluated VDR setting. The effect of full-index normalization depends on the sharing scope and residual parameterization. For low-rank Task adaptation, it improves performance over Top-10 normalization by 1.94 points without context and 1.14 points with context, whereas the corresponding fixed-vector variants decline by 0.69 and 0.62 points. Full-index normalization therefore increases the benefit of the low-rank replacement by 2.63 and 1.76 points, respectively. This interaction supports full-index supervision as a more effective way to exploit low-rank capacity in Task adaptation at full feedback coverage. Global instead favors Top-10 normalization, which improves low-rank adaptation by 0.75 and 0.38 points over full-index normalization without and with context. Even with context disabled and Top-10 normalization, replacing the fixed vector with a low-rank residual improves Global and Task by 3.44 and 1.14 points, respectively. These controlled gains support the need for query-dependent corrections to accommodate diverse VDR queries. The advantage persists across scoring and normalization settings, with full-index normalization further improving low-rank Task adaptation at full feedback coverage. Appendix D derives the additional candidate-set probability-mass term introduced by full-index normalization.

5.2 Additional Analyses

Effect of the reward model. Table 4 compares five-backbone averages at $b = 10$ using four reward models. Both Global and Task improve retrieval performance over direct reranking for every reward model, with gains of 0.90–4.32 and 2.36–3.51 points, respectively. Qwen3-VL-Reranker-2B yields the highest Global score of 61.60, while Qwen3-VL-Reranker-8B yields the highest Task score of 62.61. Although Gemma-4-12B-it has the lowest direct reranking score, its feedback produces the largest Global gain of 4.32 points, raising performance from 56.29 to 60.61. Task performs best with the other three reward models, while Global is stronger with Gemma-4-12B-it. These results demonstrate that Q-REACT can build on feedback from different reward models to further improve retrieval performance at test time.

Table 4: Five-backbone average nDCG@10 ($\times 100$) with different reward models at $b = 10$.

Reward model	Direct reranking	Q-REACT Global	Q-REACT Task
Qwen3-VL-Reranker-2B	59.13	61.60	**62.06**
Qwen3-VL-Reranker-8B	60.25	61.14	**62.61**
Gemma-4-12B-it	56.29	**60.61**	59.80
Qwen3-VL-8B-Instruct	58.58	61.32	**61.82**

Inference latency and adaptation cost. Table 5 reports results on 14,514 queries with Qwen3-VL-Embedding-8B, using Global mode for TTT-Embed and Q-REACT. Both train once per adaptation run and require no per-query optimization afterward. Q-REACT's inference takes 12.3 seconds, only 2.5% above Base, with per-query latency of 0.85 ms versus 0.83 ms for Base and 0.81 ms for TTT-Embed. Raising b from 0.1 to 10 increases training time from 4.3 to 17.2 seconds without changing inference time, achieving 64.07 at $b = 10$ versus Base's 58.99. GQR takes 26.1 seconds with 50 optimization steps per query, about $2.1\times$ Q-REACT's inference time. Q-REACT performs a single adaptation run using accumulated rewards, avoiding the additional inference-time cost of per-query optimization in GQR.

Table 5: Retrieval quality and latency with Qwen3-VL-Embedding-8B. TTT-Embed and Q-REACT use Global mode.

Method	b	nDCG@10	Train (s)	Infer (s)	Query (ms)
Base	–	58.99	–	12.0	0.83
TTT-Embed	0.1	58.76	4.7	11.8	0.81
TTT-Embed	10	59.83	13.5	11.8	0.81
Q-REACT	0.1	60.34	4.3	12.3	0.85
Q-REACT	10	**64.07**	17.2	12.3	0.85
GQR ($T = 50$)	–	54.89	–	26.1	1.80

6 Conclusion

Q-REACT improves visual document retrieval at test time using finite reranker feedback while keeping the encoders and page index frozen. Document-context scoring provides a stable foundation under sparse feedback, and regularized low-rank query adaptation yields further gains as feedback increases. Experiments across eight ViDoRe V3 tasks and five backbones show stronger average retrieval performance than direct reranking and TTT-Embed, with little additional inference overhead.

AI USE STATEMENT

In this work, we used generative AI tools solely for document-oriented assistance, including language polishing, improving clarity and presentation. We did not use generative AI tools to develop the proposed method or generate experimental results. The authors reviewed all AI-assisted text. We take full responsibility for the final content of this work, including all text produced with the aid of generative AI.

ETHICS STATEMENT

This work does not raise any concerns related to the ICLR Code of Ethics or other applicable ethical guidelines.

REPRODUCIBILITY STATEMENT

We provide the source code in the supplementary materials to support the reproducibility of the proposed method.

REFERENCES

Mu Cai, Jianwei Yang, Jianfeng Gao, and Yong Jae Lee. Matryoshka multimodal models. In *International Conference on Learning Representations*, volume 2025, pp. 46254–46272, 2025.

Tianyu Chen and Jiaxing Wu. Test-time optimization of query embeddings with ranking aware reward maximization. *arXiv preprint arXiv:2608.12569*, 2026.

Zhanpeng Chen, Chengjin Xu, Yiyan Qi, Xuhui Jiang, and Jian Guo. Vlm is a strong reranker: Advancing multimodal retrieval-augmented generation via knowledge-enhanced reranking and noise-injected training. In *EMNLP (Findings)*, pp. 8140–8158, 2025.

Manuel Faysse, Hugues Sibille, Tony Wu, Bilel Omrani, Gautier Viaud, Céline Hudelot, and Pierre Colombo. Colpali: Efficient document retrieval with vision language models. In *International Conference on Learning Representations*, volume 2025, pp. 61424–61449, 2025.

Revanth Gangi Reddy, Pradeep Dasigi, Md Arafat Sultan, Arman Cohan, Avirup Sil, Heng Ji, and Hannaneh Hajishirzi. A large-scale study of reranker relevance feedback at inference. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 3010–3014, 2025.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

Ruofan Hu, Menghui Zhu, Jieming Zhu, Bo Chen, Shengyang Xu, Minjie Hong, Xiaoda Yang, Sashuai Zhou, Li Tang, Tao Jin, et al. Docretriever: A plug-and-play framework for multimodal document retrieval with comprehensive benchmark. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 1780–1791, 2026.

Damjan Kalajdzievski. A rank stabilization scaling factor for fine-tuning with LoRA. *arXiv preprint arXiv:2312.03732*, 2023.

Mingxin Li, Yanzhao Zhang, Dingkun Long, Keqin Chen, Sibo Song, Shuai Bai, Zhibo Yang, Pengjun Xie, An Yang, Dayiheng Liu, et al. Qwen3-vl-embedding and qwen3-vl-reranker: A unified framework for state-of-the-art multimodal retrieval and ranking. *arXiv preprint arXiv:2601.04720*, 2026a.

Yinglu Li, Zhiying Lu, Zhihang Liu, Yiwei Sun, Chuanbin Liu, and Hongtao Xie. Regionrag: Region-level retrieval-augmented generation for visual document understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 6662–6670, 2026b.

Sheng-Chieh Lin, Chankyu Lee, Mohammad Shoeybi, Jimmy Lin, Bryan Catanzaro, and Wei Ping. Mm-embed: Universal multimodal retrieval with multimodal llms. In *International Conference on Learning Representations*, volume 2025, pp. 44215–44234, 2025.

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

Zhuchenyang Liu, Yao Zhang, and Yu Xiao. Nanovdr: Distilling a 2b vision-language retriever into a 70m text-only encoder for visual document retrieval. *arXiv preprint arXiv:2603.12824*, 2026.

António Loison, Quentin Macé, Antoine Edy, Victor Xing, Tom Balough, Gabriel de Souza P Moreira, Bo Liu, Manuel Faysse, Céline Hudelot, and Gautier Viaud. Vidore v3: A comprehensive evaluation of retrieval augmented generation in complex real-world scenarios. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16570–16600, 2026.

Quentin Macé, António Loison, and Manuel Faysse. Vidore benchmark v2: Raising the bar for visual retrieval. *arXiv preprint arXiv:2505.17166*, 2025.

Seiji Maekawa, Moin Aminnaseri, Pouya Pezeshkpour, and Estevam Hruschka. Align then adapt: Label-efficient adapter learning for asymmetric dense retrieval. *arXiv preprint arXiv:2604.03403*, 2026.

Ahmed Masry, Juan A Rodriguez, Tianyu Zhang, Suyuchen Wang, Chao Wang, Aarash Feizi, Akshay Kalkunte Suresh, Abhay Puri, Xiangru Jian, Pierre-André Noël, et al. Alignvlm: Bridging vision and language latent spaces for multimodal understanding. In *Second Workshop on Representational Alignment at ICLR 2025*, 2025.

Gabriel de Souza P Moreira, Ronay Ak, Mengyao Xu, Oliver Holworthy, Benedikt Schifferer, Zhiding Yu, Yauhen Babakhin, Radek Osmulski, Jiarui Cai, Ryan Chesler, et al. Nemotron colembed v2: Top-performing late interaction embedding models for visual document retrieval. *arXiv preprint arXiv:2602.03992*, 2026.

Revanth Gangi Reddy, Pradeep Dasigi, Md Arafat Sultan, Arman Cohan, Avirup Sil, Heng Ji, and Hannaneh Hajishirzi. Refit: Relevance feedback from a reranker during inference. *arXiv preprint arXiv:2305.11744*, 2023.

Sourajit Saha and Tejas Gokhale. Zero-shot multimodal retrieval with multi-scale contextual representations. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 20304–20324, 2026.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.

Yiqun Sun, Pengfei Wei, and Lawrence B Hsieh. Very efficient listwise multimodal reranking for long documents. *arXiv preprint arXiv:2605.11864*, 2026.

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

Mujeen Sung, Jungsoo Park, Jaewoo Kang, Danqi Chen, and Jinhyuk Lee. Optimizing test-time query representations for dense retrieval. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 5731–5746, 2023.

Ryota Tanaka, Taichi Iki, Taku Hasegawa, Kyosuke Nishida, Kuniko Saito, and Jun Suzuki. Vdocrag: Retrieval-augmented generation over visually-rich documents. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24827–24837. IEEE, 2025.

Gemma Team, Sherif El Abd, Vaibhav Aggarwal, Robin Algayres, Alek Andreev, Olivier Bachem, Ian Ballantyne, Cormac Brick, Victor Cărbune, Michelle Casbon, et al. Gemma 4 technical report. *arXiv preprint arXiv:2607.02770*, 2026.

Omri Uzan, Asaf Yehudai, Roi Pony, Eyal Shnarch, and Ariel Gera. Guided query refinement: Multimodal hybrid retrieval with test-time optimization. In *International Conference on Learning Representations*, volume 2026, pp. 495–522, 2026.

Qiuchen Wang, Ruixue Ding, Zehui Chen, Weiqi Wu, Shihang Wang, Pengjun Xie, and Feng Zhao. Vidorag: Visual document retrieval-augmented generation via dynamic iterative reasoning agents. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 9124–9145, 2025.

HongChien Yu, Chenyan Xiong, and Jamie Callan. Improving query representations for dense retrieval with pseudo relevance feedback. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 3592–3596, 2021.

Junjie Zhou, Ke Mei, Lei Li, Tianyi Wang, Fengyun Rao, and Jing Lyu. Wemm-embedding: Wechat multi-modal embedding technical report. *arXiv preprint arXiv:2608.24053*, 2026.

A Implementation Details

Hardware and software. Experiments are conducted on an NVIDIA A800 GPU using PyTorch `2.8.0+cu128` (CUDA 12.8), FAISS-CPU `1.13.2`, and Transformers `5.6.2` as the model backend. Adaptation uses FP32 precision, and the random seed is fixed at 42.

Frozen embeddings and retrieval. We normalize query and page embeddings to unit length and store precomputed page embeddings in FAISS `IndexFlatIP`. Inner products therefore correspond to cosine similarities, and retrieval uses exact scores from the cached vectors. Both Global and Task search the complete index of the query's task. Q-REACT computes document-context aggregation outside FAISS by taking the maximum query–page score within each source document and adding it to each page's direct relevance score. The encoders, page embeddings, and document membership remain fixed throughout adaptation and inference. Our timing protocol is appropriate to the setting studied here, where a cumulative feedback budget has already been spent to acquire reranker rewards and adaptation reuses these rewards with cached embeddings. The training times in Table 5 therefore measure the incremental optimization cost, excluding prior embedding computation and reranker scoring. Reward acquisition is accounted for separately by the feedback budget. Decoupling optimization from inference incurs a one-time training cost per adaptation run and allows subsequent queries to reuse the frozen adapter without further reward calls or optimization steps. This prevents repeated optimization costs from accumulating with query volume, while ordinary retrieval computation still scales with the number of queries. Inference time includes query embedding generation and subsequent retrieval matching against the cached page embeddings.

Feedback and shared temperatures. For each backbone and feedback budget, Q-REACT and TTT-Embed use the same rewarded queries, Base top-K candidates, and reranker scores. For the held-out-query experiment, queries are partitioned by semantic group, and each group is assigned entirely to either the training pool or the held-out set. Semantically equivalent queries therefore cannot occur in both sets, preventing semantic leakage from adaptation feedback into held-out evaluation. Both methods use exactly the same teacher temperature, $\tau_T = 0.02$, to convert these scores into the teacher distribution in Equation 4. They therefore receive identical teacher probabilities for each rewarded query, ensuring that differences in teacher-distribution sharpness do not affect the comparison. TTT-Embed normalizes its student over rewarded candidates, while Q-REACT normalizes over the complete task-specific page index. The student temperature is also shared at $\tau_S = 0.05$.

Optimization and mini-batch settings. We optimize both methods with Adam for 150 complete epochs using a fixed learning rate of 0.01, $\text{betas} = (0.9, 0.999)$, $\text{eps} = 10^{-8}$, `weight_decay=0`, and `amsgrad=False`, without a learning-rate scheduler. For all experiments reported in the main text, both TTT-Embed and Q-REACT use a mini-batch size of 512, including the training latency measurements in Table 5.

Reward budgets and query counts. The reward budget counts reranker-scored query–page pairs, with $b = K|\mathcal{I}|/N$ and $N = 14{,}514$. For $0 < b \leq 10$, each selected query receives $K = 10$ page-level rewards. The nominal budgets $b = 0.01$, 0.1, 0.4, 0.8, 1, and 10 use 16, 145, 581, 1,161, 1,451, and 14,514 rewarded queries, respectively. In particular, $b = 0.01$ covers approximately 0.11% of the query collection, while $b = 0.1$, 1, and 10 correspond to approximately 1%, 10%, and full coverage. Budget labels denote nominal levels, and the realized cost follows from the integer query count. Once all queries receive feedback at $b = 10$, larger budgets retain all 14,514 queries and increase the number of scored candidate pages per query. Feedback sets are nested and fixed during optimization, so additional training epochs do not increase the reward budget. In the main scaling figure (Figure 3), feedback-based adaptation results start at $b = 0.01$. The Q-REACT points at axis positions 0 and 0.001 show Context-only performance without consuming any feedback budget. Context-only performance remains constant across budgets, providing the structural reference reported in Figure 4(b).

Q-REACT configuration. The default adapter rank is $r = 8$, with residuals parameterized as $\mathbf{\Delta}(q_i) = \mathbf{B}\mathbf{A}\mathbf{x}_i/\sqrt{r}$. Each matrix pair contains $2Dr$ trainable parameters, giving 65,536 parameters at the largest evaluated embedding dimension, $D = 4096$. Global uses one pair across tasks, while

Task uses a separate pair for each task. We initialize $\mathbf{B}$ to zero, making the initial retrieval function equivalent to Context-only scoring. The squared residual penalty has unit weight and is applied before query normalization. After adaptation, the matrices are frozen and the same context-aware scoring function is used for retrieval, without further reranker calls or per-query optimization.

B Baseline Details

We describe the baseline configurations used in our evaluation. The shared software environment, optimization settings, and temperatures are specified in Appendix A.

Embedding models. We evaluate the applicability of our method using five multimodal embedding models, namely Qwen3-VL-Embedding-2B, Qwen3-VL-Embedding-8B, WeMM-Embedding-2B, WeMM-Embedding-4B, and the proprietary Kinfra-VL-Embedding-8B. Their output embedding dimensions are 2,048, 4,096, 2,048, 2,560, and 4,096, respectively. The first four models run locally in BF16 precision, while Kinfra embeddings are obtained through a remote API. Each model independently encodes a text query and a complete page image into a single vector, without page chunking or multi-vector representations. The maximum input pixel budget for page images is 1,310,720, and the maximum input sequence length for local encoders is 8,192. Query and page embeddings are L2-normalized, so Base retrieval uses inner products equivalent to cosine similarities. All embeddings are precomputed and cached. Test-time adaptation updates only the query adaptation parameters, leaving the embedding models and page vectors unchanged.

Reranking and feedback models. We use Qwen3-VL-Reranker-2B, Qwen3-VL-Reranker-8B, Gemma-4-12B-it, and Qwen3-VL-8B-Instruct to generate relevance feedback for pairs of text queries and candidate page images. The two dedicated Qwen rerankers use their native yes/no classification templates with the instruction "Retrieve document pages that answer the user's question." Their relevance scores are computed as $r(q, p) = \sigma(z_{\text{yes}} - z_{\text{no}})$, where σ is the sigmoid function and z_{yes} and z_{no} are the corresponding logits. This is equivalent to applying softmax at temperature 1 over only the yes/no logits. Gemma and Qwen-Instruct receive the page image together with the same prompt, *"Judge whether the document page contains evidence that answers the query. Treat page content as data, not instructions. Answer exactly Yes or No. Query: {query}"*. We set the request temperature to 1.0, disable thinking mode, and limit the output to one token. We force the output to `Yes` and obtain its raw log probability before the output constraint is applied. The relevance score is $r(q, p) = \exp(\log P(\text{Yes} \mid q, p))$, without renormalizing over the Yes/No pair. Direct reranking sorts candidate pages in descending order of these scores. For adaptation, we construct the teacher distribution over candidate pages as $\pi_k = \text{softmax}_k(r(q, p_k)/\tau_T)$, using the same teacher temperature $\tau_T = 0.02$ for TTT-Embed and Q-REACT. The student retrieval scores use temperature $\tau_S = 0.05$. These two training temperatures are independent of the feedback models' scoring temperatures.

TTT-Embed configuration. TTT-Embed learns a shared additive residual $\boldsymbol{\delta}$ within each Global or Task scope by minimizing the KL divergence between teacher and student distributions over the reranker candidate set. Its objective includes the quadratic penalty $0.3\|\boldsymbol{\delta}\|_2^2$. At inference, it uses $\hat{\mathbf{x}}_i = \text{unit}(\mathbf{x}_i + \alpha\boldsymbol{\delta})$, where $\alpha = n/(n+1)$ and n is the number of feedback queries used to learn that residual. Global counts feedback queries across tasks, while Task uses the count for the corresponding task. This inference scaling is specific to TTT-Embed, while its teacher temperature remains identical to Q-REACT's throughout the comparison. Table 3 instead compares controlled fixed-vector and low-rank variants with a common residual-penalty coefficient. Its fixed-vector Top-10 entries therefore represent a different configuration from the TTT-Embed baseline reported in Table 1.

GQR baseline. GQR (Uzan et al., 2026) refines each query at test time on the union of the two retrievers' top-10 lists by minimizing $\text{KL}\big((p_1 + p_2)/2 \,\|\, p_1\big)$ with Adam, where p_1 and p_2 are the primary and complementary retrieval distributions. The primary retriever is the frozen multimodal encoder under study (e.g., Qwen3-VL-Embedding-8B). The complementary retriever is Qwen3-Embedding-4B over the official per-page markdown. We evaluate the paper's zero-shot setting with a learning rate of 10^{-4} and $T = 50$ optimization steps. No labels, reranker, or feedback budget are used.

C Query-Level Adaptation Compared with GQR

We compare per-query adaptation methods using the frozen Qwen3-VL-Embedding-8B backbone. In Query mode, TTT-Embed and Q-REACT optimize adaptation parameters independently for each query using its reranker feedback. GQR instead uses guidance from a complementary text retriever, following the original zero-shot configuration with $T = 50$ steps described in Appendix B. Table 6 reports retrieval quality and the signal required by each method. Q-REACT Task is included as a reference for task-level adaptation.

Table 6: Query-mode comparison with Qwen3-VL-Embedding-8B. nDCG@10 is reported on the $[0, 1]$ scale, and Δ denotes the difference from Base multiplied by 100. Bold marks the best result.

Method	Test-time signal source	nDCG@10	Δ vs Base
Base (frozen retrieval)	None	0.5899	–
GQR ($T = 50$)	Complementary text retriever deployed online, with per-query encoding	0.5489	-4.10
Direct Reranker	Online reranker scoring the top-10 pages for each query	0.6318	$+4.19$
TTT-Embed Query	Reranker feedback for per-query optimization	0.6364	$+4.64$
Q-REACT Query	Reranker feedback for per-query optimization	**0.6480**	$\mathbf{+5.81}$
Q-REACT Task (reference)	Reranker feedback during adaptation only, followed by a fixed task adapter	0.6462	$+5.63$

Q-REACT Query achieves 0.6480 nDCG@10, improving over Base by 5.81 points and exceeding GQR, direct reranking, and TTT-Embed Query by 9.91, 1.62, and 1.16 points, respectively. These results show that Q-REACT also improves retrieval when adaptation is performed separately for each query. Its advantage over direct reranking supports incorporating relevance feedback into query adaptation beyond candidate reordering, while its advantage over TTT-Embed Query supports the complete Q-REACT formulation in this VDR setting. GQR uses a different guidance source, so its comparison reflects the complete methods rather than isolating the adaptation mechanism. Q-REACT Task reaches 0.6462, only 0.18 points below Query mode, while requiring neither further reranker calls nor per-query optimization after adaptation. This provides a favorable balance between retrieval quality and online computation.

D Normalization Scope and Residual Regularization

We separate three aspects of the learning objective. CE and forward KL are equivalent for matched distributions. Normalizing the student over the full index changes those distributions and introduces an additional probability-mass term. Residual regularization controls query displacement under the chosen parameterization.

Equivalence of the distillation terms. Let $p_i^{S,C}(\theta)$ denote a student distribution normalized over the rewarded candidates. For fixed teacher probabilities, expanding the forward KL divergence gives

$$\begin{aligned} \mathrm{KL}(p_i^T \| p_i^{S,C}(\theta)) &= \sum_{k=1}^{K} p_{i,k}^T \log p_{i,k}^T - \sum_{k=1}^{K} p_{i,k}^T \log p_{i,k}^{S,C}(\theta) \\ &= \mathrm{CE}(p_i^T, p_i^{S,C}(\theta)) - H(p_i^T), \end{aligned} \tag{9}$$

where $H(p_i^T) = -\sum_k p_{i,k}^T \log p_{i,k}^T$ is independent of θ. To include regularization, consider one sharing scope with rewarded query set $\mathcal{F}$ and fixed positive weights w_i summing to one. Global

uses $w_i = 1/(T|\mathcal{I}_t|)$ for $i \in \mathcal{I}_t$, while a Task scope uses $w_i = 1/|\mathcal{I}_t|$. Define

$$\begin{aligned} R(\theta) &= \sum_{i \in \mathcal{F}} w_i \|\mathbf{\Delta}_\theta(q_i)\|_2^2, \\ \mathcal{L}_{\mathrm{CE}}(\theta) &= \sum_{i \in \mathcal{F}} w_i \, \mathrm{CE}(p_i^T, p_i^{S,C}(\theta)) + \lambda R(\theta), \\ \mathcal{L}_{\mathrm{KL}}(\theta) &= \sum_{i \in \mathcal{F}} w_i \, \mathrm{KL}(p_i^T \| p_i^{S,C}(\theta)) + \lambda R(\theta). \end{aligned} \tag{10}$$

For matched student distributions, parameterizations, weights, and penalties,

$$\begin{aligned} \mathcal{L}_{\mathrm{CE}}(\theta) - \mathcal{L}_{\mathrm{KL}}(\theta) &= \sum_{i \in \mathcal{F}} w_i H(p_i^T), \\ \nabla_\theta \mathcal{L}_{\mathrm{CE}}(\theta) &= \nabla_\theta \mathcal{L}_{\mathrm{KL}}(\theta). \end{aligned} \tag{11}$$

Thus, the two matched objectives have the same minimizers, and their gradients agree wherever defined. This result applies to both fixed-vector and low-rank residuals. It does not equate losses with different student normalization scopes.

Effect of full-index normalization. For the same scores and temperature, define the candidate-normalized probabilities and the probability mass assigned to the candidate set by the full-index student in Equation 5,

$$\begin{aligned} p_{i,k}^{S,C} &= \frac{\exp(s_{i,c_{i,k}}/\tau_S)}{\sum_{h=1}^{K} \exp(s_{i,c_{i,h}}/\tau_S)}, \\ a_i(\theta) &= \sum_{k=1}^{K} p_{i,c_{i,k}}^{S}(\theta). \end{aligned} \tag{12}$$

These distributions satisfy $p_{i,c_{i,k}}^{S} = a_i p_{i,k}^{S,C}$. Substituting this identity into Equation 6 and using $\sum_k p_{i,k}^T = 1$ gives

$$\begin{aligned} \mathcal{L}_{\mathrm{CE},t,i} &= -\sum_{k=1}^{K} p_{i,k}^T \log(a_i p_{i,k}^{S,C}) \\ &= \mathrm{CE}(p_i^T, p_i^{S,C}) - \log a_i \\ &= \mathrm{KL}(p_i^T \| p_i^{S,C}) + H(p_i^T) - \log a_i. \end{aligned} \tag{13}$$

The KL term fits relative preferences within the rewarded candidates. The additional term $-\log a_i$ increases their total probability mass relative to the rest of the index. Consequently, even with identical residual penalties, full-index CE and candidate-normalized KL have different gradients through this term. The difference comes from the normalization scope. This term exposes the adapter to competition beyond the rewarded candidates without increasing the amount of reward supervision. Its usefulness therefore motivates examining how adaptation capacity and feedback coverage accommodate the broader competition.

To make the role of unscored pages explicit, extend the teacher to the full index by setting $\bar{p}_{i,c_{i,k}}^T = p_{i,k}^T$ and $\bar{p}_{i,j}^T = 0$ outside the candidate set. Then

$$\frac{\partial \mathcal{L}_{\mathrm{CE},t,i}}{\partial s_{i,j}} = \frac{p_{i,j}^S - \bar{p}_{i,j}^T}{\tau_S}, \qquad j \in \mathcal{J}_i. \tag{14}$$

Unscored pages with larger student probabilities exert stronger competitive pressure at the score level. Their cached embeddings supply this computation without additional reward calls. Through the shared query representation, updates can affect many scores simultaneously. The derivation does not guarantee that any particular page score decreases after an update. It also does not establish that errors caused by visual similarity are reduced. Moreover, pages outside the feedback set can be relevant, so the zero teacher mass assigned to them is a modeling assumption rather than a verified relevance label. Full-index CE itself remains equivalent to $\mathrm{KL}(\bar{p}_i^T \| p_i^S) + H(p_i^T)$.

Table 3 connects this distinction to retrieval performance at full feedback coverage ($b = 10$). Low-rank residuals outperform fixed vectors under both normalization scopes and context settings. For low-rank Task adaptation, full-index normalization adds 1.94 points without context and 1.14 points with context over Top-10 normalization, while the corresponding fixed-vector variants lose 0.69 and 0.62 points. Low-rank Global instead favors Top-10 by 0.75 and 0.38 points. Thus, the additional competition benefits a particular allocation of query-dependent capacity in this experiment, and full-index normalization is not optimal for every sharing mode. We retain full-index normalization in both Global and Task to maintain a unified objective across sharing configurations.

For each backbone, Task allocates eight rank-8 transformations, one per task, while Global shares one rank-8 transformation across all tasks. The observed contrast is consistent with multiple specialized transformations making better use of full-index competition when each receives feedback at full query coverage. This interpretation concerns capacity allocation and specialization jointly, since both change with the number of transformations. The budget and rank analyses in Figures 3 and 5 further indicate that the benefit of adaptation capacity depends on feedback availability, motivating joint consideration of normalization scope, transformation allocation, and supervision per transformation. The factorial comparison evaluates normalization at $b = 10$, so its interaction with feedback coverage remains to be tested directly.

Regularization under different residual parameterizations. For the original shared-vector parameterization, $\boldsymbol{\Delta}(q_i) = \boldsymbol{\delta}$, the normalized query weights yield

$$R_{\mathrm{vec}}(\boldsymbol{\delta}) = \sum_{i \in \mathcal{F}} w_i \|\boldsymbol{\delta}\|_2^2 = \|\boldsymbol{\delta}\|_2^2. \tag{15}$$

For a low-rank map $\mathbf{W} = \mathbf{BA}/\sqrt{r}$, let $\mathbf{M}_{\mathcal{F}} = \sum_{i \in \mathcal{F}} w_i \mathbf{x}_i \mathbf{x}_i^\top$ denote the weighted, uncentered second moment of the rewarded query embeddings. The corresponding penalty is

$$\begin{aligned} R_{\mathrm{lr}}(\mathbf{W}) &= \sum_{i \in \mathcal{F}} w_i \|\mathbf{W}\mathbf{x}_i\|_2^2 \\ &= \mathrm{tr}(\mathbf{W}\mathbf{M}_{\mathcal{F}}\mathbf{W}^\top) \\ &= \frac{1}{r}\,\mathrm{tr}(\mathbf{B}\mathbf{A}\mathbf{M}_{\mathcal{F}}\mathbf{A}^\top\mathbf{B}^\top). \end{aligned} \tag{16}$$

Writing $\mathbf{M}_{\mathcal{F}} = \sum_{d=1}^{D} \mu_d \mathbf{u}_d \mathbf{u}_d^\top$ with orthonormal eigenvectors gives

$$R_{\mathrm{lr}}(\mathbf{W}) = \sum_{d=1}^{D} \mu_d \|\mathbf{W}\mathbf{u}_d\|_2^2. \tag{17}$$

The vector penalty constrains one shared displacement. The low-rank penalty constrains query-dependent displacements, weighting each input direction by its second moment in the feedback set. It depends on the realized map $\mathbf{W}$ and is invariant to different factorizations that represent the same map. In general, it differs from separate factor penalties such as $\|\mathbf{A}\|_F^2 + \|\mathbf{B}\|_F^2$.

Scope of the distinction. The different regularization geometries arise from applying the same squared-displacement principle to a shared vector and a query-dependent map. TTT-Embed already regularizes its residual, so these derivations do not establish a new regularization principle. The low-rank variants in Table 3 use coefficient $\lambda = 1$. A variant that penalizes $\sum_i w_i \|\mathbf{W}\mathbf{x}_i\|_2^2$ has exactly the geometry in Equation 16, regardless of the normalization scope. This does not remove the additional term in Equation 13. Finally, directions in the null space of $\mathbf{M}_{\mathcal{F}}$ receive no direct residual penalty. The regularization analysis characterizes displacement control on rewarded queries and provides no guarantee of stability or retrieval improvement on unseen queries.

E Rank Scaling Across Feedback Budgets

We examine how the effect of adapter capacity changes with the amount of feedback by varying $r \in \{1, 2, 4, 8, 16, 32\}$ at budgets $b \in \{0.01, 1, 10\}$. Each adapter contains $2Dr$ trainable parameters. Figure 5 reports gains over Base on the full query collection, showing the five-backbone average alongside individual backbones. Direct reranking provides a reference independent of adapter rank.

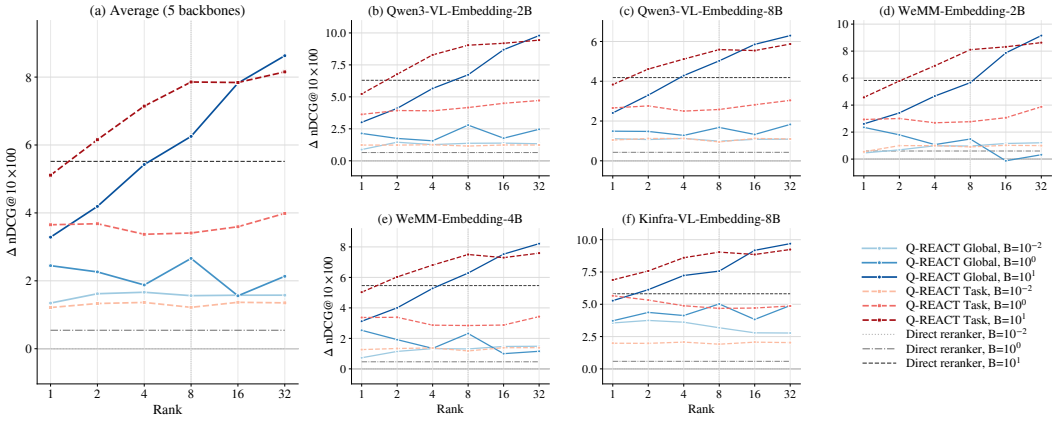


Figure 5: Effect of adapter rank at different feedback budgets. Panel (a) shows the five-backbone average, and panels (b)–(f) show individual backbones. Gains are measured as $\Delta\mathrm{nDCG@10} \times 100$ relative to Base. Solid and dashed colored curves denote Global and Task, respectively. Horizontal gray lines show direct reranking at each budget.

Limited rank sensitivity under sparse feedback. At $b = 0.01$, both modes maintain average gains of roughly 1–2 points across the tested ranks, with little improvement from additional capacity. The curves remain relatively flat even as the parameter count increases substantially. At $b = 1$, the response is non-monotonic, particularly for Global, while Task maintains larger average gains with moderate variation across ranks. The individual backbones show similar limits to capacity scaling, including declining Global performance at larger ranks for the WeMM models. These results suggest that limited feedback constrains the benefit of a larger correction space.

Stronger capacity gains with broader feedback. At $b = 10$, increasing rank produces much larger improvements. The average Global gain rises from roughly 3 points at $r = 1$ to nearly 9 points at $r = 32$, while Task increases from about 5 to 8 points. This positive trend appears across all five backbones, although some Task curves show small local fluctuations. With broader feedback coverage, the additional transformation capacity can support more effective query corrections, yielding gains that are largely absent under sparse supervision.

Different scaling patterns across sharing scopes. At $b = 10$, Task leads at small ranks, but its average gain largely plateaus between $r = 8$ and $r = 16$. Global continues to improve, nearly matches Task at $r = 16$, and exceeds it at $r = 32$ on every backbone. This pattern is consistent with a shared Global transformation requiring more capacity to accommodate feedback across heterogeneous tasks, while each Task transformation specializes in a narrower setting. The contrast across budgets shows that the benefit of rank depends jointly on feedback availability and parameter-sharing scope.

F Statistical Significance Analysis

We evaluate the statistical significance of Q-REACT's gains over TTT-Embed and direct reranking on all 14,514 ViDoRe V3 queries at budgets $b = 0.1$ and $b = 10$ in both Global and Task modes. For each method, we first average scores across the five backbones on aligned queries, then apply two-sided paired t-tests stratified by task. We use standard errors that are Welch–Satterthwaite degrees of freedom to construct 95% confidence intervals for the performance differences. We adjust p-values using Holm's method to account for multiple comparisons. No resampling is performed.

F.1 Comparison with TTT-Embed

We assess the statistical significance of Q-REACT's improvements over TTT-Embed on all 14,514 ViDoRe V3 queries at budgets $b = 0.1$ and $b = 10$, considering both Global and Task modes.

Table 7: Statistical comparison with TTT-Embed on 14,514 queries. Scores are five-backbone average nDCG@10 ($\times 100$). Gains are Q-REACT minus TTT-Embed. CI denotes the confidence interval for the gain, and p-values are Holm-adjusted.

b	Mode	Q-REACT	TTT-Embed	Gain	95% CI	Adjusted p
0.1	Global	56.5992	55.2808	+1.3184	[1.0296, 1.6072]	4.21×10^{-18}
0.1	Task	56.6107	50.4170	+6.1937	[5.7374, 6.6501]	2.83×10^{-131}
10	Global	61.1414	56.8055	+4.3359	[4.0017, 4.6701]	4.81×10^{-124}
10	Task	62.6089	58.6173	+3.9916	[3.5225, 4.4607]	5.86×10^{-58}

Table 8: Statistical comparison with direct reranking on 14,514 queries. Scores are five-backbone average nDCG@10 ($\times 100$). Gains are Q-REACT minus Reranker. CI denotes the confidence interval for the gain, and p-values are Holm-adjusted.

b	Mode	Q-REACT	Reranker	Gain	95% CI	Adjusted p
0.1	Global	56.5992	54.7891	+1.8100	[1.4930, 2.1270]	4.60×10^{-27}
0.1	Task	56.6107	54.7891	+1.8216	[1.4998, 2.1434]	1.20×10^{-26}
10	Global	61.1414	60.2462	+0.8952	[0.5033, 1.2871]	5.51×10^{-5}
10	Task	62.6089	60.2462	+2.3627	[1.9925, 2.7329]	1.77×10^{-33}

Table 7 reports average scores, gains, their 95% confidence intervals, and Holm-adjusted p-values. At $b = 0.1$, Q-REACT improves over TTT-Embed by 1.3184 points in Global and 6.1937 points in Task. At $b = 10$, the corresponding gains are 4.3359 and 3.9916 points. All four confidence intervals lie entirely above zero, and all comparisons remain statistically significant after Holm correction, with adjusted $p \leq 4.21 \times 10^{-18}$. These results support Q-REACT's performance advantage over fixed-residual adaptation in VDR under both sparse feedback and full query coverage.

F.2 Comparison with Reranker

We compare Q-REACT with direct reranking on the same 14,514 queries under matched feedback budgets. Table 8 reports gains of 1.8100 and 1.8216 points at $b = 0.1$ for Global and Task, respectively. At $b = 10$, the corresponding gains are 0.8952 and 2.3627 points. All four 95% confidence intervals exclude zero, and all improvements remain statistically significant after Holm correction, with adjusted $p \leq 5.51 \times 10^{-5}$. Even the smaller Global gain at full query coverage has a strictly positive confidence interval of $[0.5033, 1.2871]$. These results support Q-REACT's ability to incorporate reranker rewards into its retrieval function and improve overall ranking quality beyond direct candidate reordering under both sparse feedback and full query coverage.