GRAD: Generalizing RAG Adaptation with Decoding

Youngwon Lee[1]* **Seung-won Hwang**[1]† **Zhewei Yao**[2] **Yuxiong He**[2]
[1]Seoul National University [2]Snowflake AI Research

Abstract

Retrieval-augmented generation needs generation to follow retrieved evidence across shifting domains and prompt layouts, but training a new stronger model per task is costly. To this end, we propose GRAD, an adaptive *decoding-time* framework that keeps the base generator fixed and composes small, objective-specific guidance at inference. A key advantage of this design is enabling mix and match diverse RAG objectives: *model scaling* (MS), *domain adaptation* (DA) and *positional debiasing* (DB) can be integrated as token-level guidance terms, and new objectives can be easily plugged in. Across public benchmarks and private settings with *no in-domain* labels, GRAD improves accuracy with favorable latency, offering strong trade-offs versus scaling while reliably activating helpful objectives and suppressing harmful ones, adaptively to tasks.

1 Introduction

This paper studies how to steer Retrieval-Augmented Generation (RAG) objectives without retraining large language models (LLMs). Early steering work showed that one can shape a base model's next-token distribution at inference time. To illustrate, *Proxy Tuning* (Liu et al., 2024a), without finetuning a large model M into M', mimics its effect by finetuning smaller helper models m into m'. Instead of directly finetuning $M \to M'$, it mimics the behavior difference between smaller counterparts, which has been reported as prior work for approximating narrow finetuning objectives with well-supervised behaviors, such as toxicity reduction or stylistic control.

Our goal is to generalize steering for RAG adaptation objectives. Given a base model, there are well-known adaptation goals: **domain adaptation** when retrieved passages contain specialized knowledge that base model may not already possess, and **positional debiasing** when long contexts include distractors or when key evidence appears later in the sequence. These objectives often interact and possibly conflict with each other. Consequently, training a single specialized model M' to represent the "right" mixture of competing objectives for every scenario becomes impractical.

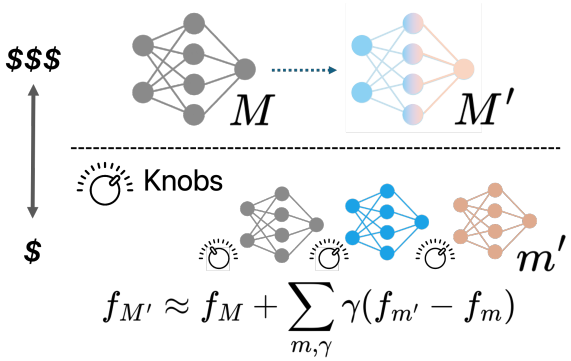

Figure 1: GRAD replaces monolithic retraining of M' by aggregating smaller models. Task-level adaptation knobs turn on useful signals while turning off harmful ones. This allows to adapt to different tasks, in which a specific objective may help or hurt.

We thus take a different view: instead of searching for a single, task-specific M', we prepare a small set of steering objectives implemented as helper models that produce token-level guidance scores, and 'knobs' γ's to adapt the right mixture for the given scenario. This shifts the problem from training a monolithic model for each specific task into (a) building a set of helper models and (b) integrating with knobs.

To this end, we present GRAD (Generalizing Retrieval-augmented generation Adaptation with Decoding), a unified decoding-time framework for RAG that adaptively activates objectives per task. We instantiate GRAD with three widely used RAG objectives, **Model Scaling (MS)**, **Domain Adap-**

* Work done while at Snowflake.
† Correspondence to: seungwonh@snu.ac.kr.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 45274–45290

tation (DA), and **Positional Debiasing (DB)**, by converting each into a token-level guidance score. Small helper models supply *weak-to-strong* transfer signals (Burns et al., 2024), while the base generator M is left unchanged. Our adaptation suppresses guidance that conflicts with the base distribution, yielding stable behavior even without in-domain training data.

Figure 1 describes combining objectives at decoding time. Towards the goal of approximating M' with mixed objectives (colors), we use an *adaptive mix* of small helper models, corresponding to MS (gray), DA (blue), and DB (apricot): Each helper supplies a token-level guidance term, and task-level activation selects a conflict-free subset, adaptively to task (Section 3.4). For brevity, here we illustrate with binary knobs, where each objective is either activated ($\gamma = 1$) or deactivated ($\gamma = 0$); we optimize for arbitrary weights in Section 3.4.

We first note that GRAD provides a unifying perspective on existing decoding methods: for example, activating DA only, using γ's $= (0, 1, 0)$, recovers proxy tuning as a special case. Next, as shown later empirically (Table 2 and 3), depending on tasks (benchmarks), GRAD chooses the optimal adaptive combination. For example, it chooses to activate all objectives (using γ's $= (1, 1, 1)$) for HotpotQA(HQA), while activating the DB objective only (using γ's $= (0, 0, 1)$) for private RAG settings where perfect in-domain data is sensitive and/or unavailable (PQ).

Finally, GRAD's design is also naturally extensible, where new objectives can be easily plugged in, as illustrated in Appendix H.

In summary, our contributions are as follows:

- We propose GRAD, an *adaptive* guided decoding framework that selects and combines the helper models per task instead of training a new M'.
- We introduce CCD, a decoding objective that disentangles positional debiasing from domain adaptation.
- We show *extensibility*: New objectives can be integrated seamlessly as needed.
- We demonstrate robust generalization across public and private RAG scenarios, with favorable accuracy-latency trade-offs.[1]

[1]Code at https://github.com/ludaya/grad.

2 Related Work

Domain adaptation in RAG RAFT (Zhang et al., 2024) optimizes language models for domain-specific RAG by jointly adapting retrieval and generation to better reflect the target domain's characteristics, while other works jointly update the retriever as well (Siriwardhana et al., 2023; Mao et al., 2024). However, end-to-end training incurs higher cost while making the system monolithic, hard to be combined with other RAG objectives.

Position bias in RAG Position bias, often termed the "lost-in-the-middle" problem (Liu et al., 2024b), occurs when important segments of retrieved passages receive less attention from the model. Existing solutions fall into train-time and inference-time approaches.

One inference-time strategy, self-consistency (Wang et al., 2023), reduces bias by aggregating outputs across multiple input orderings, but at a high inference cost. More efficient meta-generation (Lee et al., 2025b) minimizes redundancy and inference cost. Other inference-time strategies include Hsieh et al. (2024), which tracked the average attention weights to calibrate the effect of position bias.

Decoding objectives Contrastive decoding (Li et al., 2023) leverages the difference between expert and amateur model distributions to guide generation. Similar line of work has explored various contrastive objectives that target a specific aspect of generation quality, such as controlling toxicity of the generated text (Cheng et al., 2023; Liu et al., 2024a). While not directly altering the decoding objective, speculative decoding (Leviathan et al., 2023) also utilizes smaller auxiliary model to accelerate generation.

In a broader perspective, these methods can be interpreted as an instantiation of test-time reward-guided text generation (Khanov et al., 2024; Xu et al., 2025), where the logit difference serves as an implicit reward model over the (unfinished) sequences.

Weak-to-strong generalization Weak-to-strong generalization refers to distilling predictions of a weak teacher into a strong student model (Burns et al., 2024). Many works showed, against the doubt, weak models are effective in refining labels for strong model (Somerstep et al., 2025) or contrastive scoring samples from the strong (Zhou et al., 2024). Our proposed decoding objectives

that contrast weak models' logits to guide a strong model are also considered weak-to-strong (Fan et al., 2024).

Our distinction Our distinction is unifying RAG objectives, without costly large-model updates, realizing weak-to-strong generalization by activating only the objectives that benefit each task. GRAD, by aggregating seemingly weak contrastive signals from 1B models, can effectively improve much larger models (8B and 70B).

3 Method

3.1 Decoding Objectives

We begin by introducing basic notations and formally describing the token-level decoding objectives. An autoregressive language model has been typically trained to maximize the data likelihood, learning to predict the next token probability given the prefix of token sequences. The model achieves this by calculating a probability distribution over possible next tokens. The output probability distribution at each time step t is typically obtained by applying softmax to raw outputs, or logits,

$$p(\cdot \mid y_{<t}) = \text{softmax}(f_{\theta,t}), \tag{1}$$

where $f_{\theta,t} = [f(y_t \mid y_{<t}; \theta)]_{y_t \in \mathcal{V}}$ represents the logits produced by the model θ for each token in the vocabulary $\mathcal{V}$. While it is widespread to empirically adjust the sharpness of the distribution with a temperature T as follows,

$$p(\cdot \mid y_{<t}) = \text{softmax}(f_{\theta,t}/T), \tag{2}$$

we omit T for presentation brevity.

Decoding sequences from these probabilities involves sampling strategies, such as unbiased sampling or approximate MAP decoding (e.g., greedy or beam search), all of which aim to recover the most desirable sequence from probabilities.

3.2 Guided Decoding

Alternatively, external reward, or guidance terms R can explicitly guide the generation process, towards optimizing a target objective. A generalized form of this approach from external guidance score R_t can be expressed as

$$p(\cdot \mid y_{<t}) = \text{softmax}(f_t + \gamma R_t), \tag{3}$$

where γ is a hyperparameter controlling the influence of the external signal on shifting the base logits from the LLM at time step t, f_t. Then, the combined score is converted to probability distribution over the vocabulary $\mathcal{V}$ by applying softmax. However, training token-level reward, compared to trajectory-level is known to be tricky, due to sparse supervisory signals (Xu et al., 2025; Rashid et al., 2025).

3.3 GRAD: Generalizing RAG with Decoding

Instead of training a reward model that outputs R_t at each time step t, we derive guidance signals from logit differences between models to contrast:

$$R_t = f_t^+ - f_t^-. \tag{4}$$

With this general template, we describe how the three RAG strategies, **MS**, **DA** and **DB** can be transferred and unified into decoding objectives as below.

MS Contrastive decoding (CD) contrasts predictions from a larger model θ^l and a smaller model θ^s, to extrapolate towards a hypothetical infinite-sized model θ^∞. The contrastive logits for computing the guidance terms in Eq. 4 are given as

$$\begin{aligned} f_t^+ &= f\left(y_t \,\middle|\, y_{<t}; \theta^l\right), \\ f_t^- &= f\left(y_t \mid y_{<t}; \theta^s\right). \end{aligned} \tag{5}$$

In our setting, θ^l is set as the base LLM, θ, while θ^s is chosen as a smaller model from the same model family. As outlined earlier CD objective of GRAD in Eq. 5 aims to simulate a hypothetical large LLM by extrapolating the base model's prediction away from that of the small model's. This formulation is slightly different from the original definition by Li et al. (2023), which is discussed in more detail in Appendix B.

DA Proxy tuning (Liu et al., 2024a) identifies and transfers the benefits of finetuning using small models, instead of directly finetuning a large one:

$$\begin{aligned} f_t^+ &= f\left(y_t \mid y_{<t}; \theta^s_{\text{ft}}\right), \\ f_t^- &= f\left(y_t \mid y_{<t}; \theta^s\right), \end{aligned} \tag{6}$$

where θ^s is a small base model, and θ^s_{ft} is its finetuned version. While prior work focused on surface-level attributes like toxicity (Liu et al., 2021), our approach captures and transfers for RAG adaptation (Zhang et al., 2024).

In private, domain-specific RAG settings, a well-aligned training dataset $\mathcal{D}$ for DA that matches the test distribution can be absent. We contrast and show robustness in such scenarios in later sections.

DB Here, we elaborate on how the proposed Consistency Contrastive Decoding (CCD) debiases, or minimizes shifts in the model's output distribution caused by changes in input context arrangement. Existing test-time methods require multiple forward passes with perturbed inputs (Wang et al., 2023; Lee et al., 2025b). Literal translation of such approach would be contrasting logits from two same sets of input contexts arranged in different order, x^+ and x^- as follows:

$$\begin{aligned} f_t^+ &= f\left(y_t \mid x^+, y_{<t}; \theta\right), \\ f_t^- &= f\left(y_t \mid x^-, y_{<t}; \theta\right). \end{aligned} \tag{7}$$

Methods such as instructive decoding (Kim et al., 2024) share the above formulation in Eq. 7, where the two inputs correspond to prompting the LLM with positive and negative task instructions. For debiasing purposes, selecting a definitive positive-negative ordering is nontrivial, and accounting for all $n!$ permutations incurs significant test-time overhead.

Instead, we shift debiasing effort to training a small model θ^s_{db} using CORD (Lee et al., 2025a), which penalizes inconsistencies in model outputs given the same input contexts ordered differently with a train-time consistency loss. Intuitively, we extrapolate from biased model predictions toward less biased ones. To this end, two finetuned models—one trained with consistency and one without—are contrasted to isolate the effect of debiasing via input perturbations, hence the name Consistency Contrastive Decoding.

$$\begin{aligned} f_t^+ &= f\left(y_t \mid x, y_{<t}; \theta^s_{\text{db}}\right), \\ f_t^- &= f\left(y_t \mid x, y_{<t}; \theta^s_{\text{ft}}\right). \end{aligned} \tag{8}$$

Unlike DA, this consistency-aware approach generalizes well across datasets. By contrasting logits from trained models as in Eq. 8, CCD removes dataset-specific biases from finetuning, improving generalization to new domains and tasks.

Combined: GRAD Finally, GRAD combines all three RAG strategies expressed as decoding objectives,

$$p(\cdot \mid y_{<t}) = \text{softmax}\left(f_t + \sum_i \gamma_i R_{i,t}\right), \tag{9}$$

where the token-level reward terms $R_{\text{MS},t}$, $R_{\text{DA},t}$, and $R_{\text{DB},t}$ are defined with Eq. 4, contrasting model logits from Eq. 5, 6, and 8, respectively. This can be also seen in Algorithm 1.

In the case of single objective optimization as in Eq. 3, γ has been typically determined empirically, choosing the value of γ that maximizes validation performance. For multi-objective optimization as in GRAD, it has been widespread to assume that relative importance of each objective known a priori (Shi et al., 2024). We argue such assumption is impractical in RAG, and propose *task-level adaptation* that selects objective activations and strengths without any prior information, as in the next section.

3.4 Task-Level Adaptation

While an objective that helps in one setting may hurt in another, optimal weights are not known a priori. We therefore propose a mechanism that *adapts* each objective to the current task scenario by automatically selecting γ_i, the weight of objective i for the given task.

Decomposing magnitude and activation For each objective $i \in \{\text{DA}, \text{DB}, \text{MS}\}$, we decompose its effective decoding weight as

$$\gamma_i = \gamma_i^0 \cdot \gamma_i^1, \tag{10}$$

where γ_i^0 controls the magnitude of the objective and $\gamma_i^1 \in \{0, 1\}$ is a task-level activation switch, controlling whether the objective is applied.

When labeled validation data is accessible Prior decoding-time optimization methods typically assume a single objective and select the guidance strength γ by labeled validation accuracy (e.g., contrastive decoding and proxy tuning). If a labeled validation split is available for the target scenario, the simplest procedure follows standard practice: we choose $\gamma_i^0 \in \Gamma$ by downstream validation accuracy, and activate objective i if it improves validation performance compared to not using it (conditioned on any previously activated objectives). This provides a straightforward label-based instantiation of task-level adaptation.

When labeled validation data is not accessible Meanwhile, many private or rapidly changing RAG settings do not have access to a labeled validation split with ground-truth answers for tuning; in such cases, we require a *label-free* signal to decide both γ_i^0 and γ_i^1.

We build on the divergence-alignment criterion of Fan et al. (2024), which assesses whether apply-

ing a guidance objective induces a distribution shift whose *scale* is compatible with the shift captured by the objective's expert–amateur contrast. Let p denote the current base next-token distribution (including any previously activated objectives), and let p' denote the guided distribution when applying objective i with magnitude γ_i^0 at decoding step t. Let (q_i, q_i') be the amateur and expert distributions defining objective i (Eq. 4):

$$q_i' = \text{softmax}(f_{i,t}^+),\ q_i = \text{softmax}(f_{i,t}^-). \quad (11)$$

We define the token-level alignment signal

$$d_t(\gamma_i^0) = \Big(\big(\text{KL}(p \,\|\, p') + \text{KL}(p' \,\|\, p)\big) - \big(\text{KL}(q_i \,\|\, q_i') + \text{KL}(q_i' \,\|\, q_i)\big)\Big)^2. \quad (12)$$

Small $d_t(\gamma_i^0)$ indicates that the guided shift $p \to p'$ has a similar magnitude to the expert–amateur shift $q_i \to q_i'$, suggesting that objective i can be applied without inducing an overly incompatible change to the base model at that decoding step.

As we make a *single* decision per task scenario (e.g., a dataset or deployment configuration), we aggregate these token-level signals, by examining how often the alignment condition holds, avoiding the need to adapt at every decoding step. That is, by activating objective i only if there exists a magnitude for which the alignment guard holds frequently enough; otherwise we suppress the objective for that task scenario.

For a candidate magnitude $\gamma \in \Gamma$, we compute the proportion of decoding steps satisfying the guard $\mathbb{I}[d_t(\gamma) < \tau]$, averaged over time and a small unlabeled sample set S of queries:

$$A(\gamma) = \mathbb{E}_{x \sim S}\left[\frac{1}{T_x}\sum_{t=1}^{T_x}\mathbb{I}\big[d_t(\gamma) < \tau\big]\right]. \quad (13)$$

We then select the magnitude and activation as follows:

$$\gamma_i^0 = \arg\max_{\gamma \in \Gamma} A(\gamma),\ \gamma_i^1 = \mathbb{I}\left[\max_{\gamma \in \Gamma} A(\gamma) \geq \rho\right]. \quad (14)$$

Accounting for objective interactions. Objectives can exhibit synergy or interference; in order to capture interactions without enumerating all combinations, we evaluate objectives sequentially, conditioning each decision on the set of previously activated objectives by treating their contributions as part of the current base distribution p. We adopt the fixed order of DA first, followed by DB then MS, based on the relative specificity of the objectives. DA relies on task- and domain-specific supervision and therefore has the most direct impact on correctness. DB addresses structural biases in retrieval (e.g., position effects) that are largely task-agnostic but should not override domain alignment. MS reflects model capacity differences and is the most global signal, making it safest to apply after the other objectives.

Algorithm 2 formalizes our adaptation mechanism, for both scenarios (when validation accuracy is available and when it is not).

4 Results

4.1 Experimental Settings

We validate the effectiveness of GRAD on the following diverse RAG scenarios: For public benchmarks, we consider **MARCO** (Bajaj et al., 2018) for single-hop question answering and **HotpotQA** (Yang et al., 2018), or HQA, for multi-hop question answering, both of which have well-aligned training data available.

We deploy it in a real-life proprietary enterprise setting, which we denote as **PQ**, on a private corpus of company internal PDFs unseen from LLM, with no training data available. For reproduction, we also consider **NQ** (NaturalQuestions, Kwiatkowski et al., 2019) with no training data.

While MARCO and HQA have well-defined in-domain training sets,[2] both NQ and PQ lack such training data. However, NQ builds on the Wikipedia corpus as MARCO and HQA, their training data can be transferred for NQ, and the LLM has likely seen the data during pretraining. Thus, we intend these two scenarios, NQ and PQ, to analyze the impact of different degrees of finetuning misalignment, which is a frequently faced challenge of enterprise RAG serving domain-specific or proprietary knowledge. Also, for PQ, conversion process from PDF to text introduces realistic noise in practical RAG scenarios, while further increasing the divergence from standard training distributions.

Unless otherwise noted, the base LLM is an 8B LLaMA 3.1 model and each SLM used for decoding objectives is a 1B LLaMA 3.2 model. We

[2] HQA has a dedicated training set, whereas we split the development set of MARCO and repurpose the held-out set for training.

Algorithm 1 GRAD: Turning RAG Objectives to Token-level Decoding Objectives

Require: Input x, Base LLM θ, SLM θ^s, Domain-specific Training Set $\mathcal{D}$
Ensure: Guided output y

$\theta^s_{\text{ft}} \leftarrow$ Finetuned SLM on $\mathcal{D}$
$\theta^s_{\text{db}} \leftarrow$ Finetuned SLM on $\mathcal{D}$ with consistency objective for debiasing
$t \leftarrow 1$
while y does not meet stopping criteria **do**
  $f_{\text{base}} \leftarrow f(x, y_{<t}; \theta)$ ▷ Logits from the base LLM
  $f_{\text{s}} \leftarrow f(x, y_{<t}; \theta^s)$ ▷ Logits from the base SLM
  $f_{\text{ft}} \leftarrow f(x, y_{<t}; \theta^s_{\text{ft}})$ ▷ Logits from the finetuned SLM
  $f_{\text{db}} \leftarrow f(x, y_{<t}; \theta^s_{\text{db}})$ ▷ Logits from the finetuned SLM with consistency-aware training
  $f_t \leftarrow f_{\text{base}} + \gamma_{\text{MS}} \cdot \underbrace{(f_{\text{base}} - f_{\text{s}})}_{\text{MS objective}} + \gamma_{\text{DA}} \cdot \underbrace{(f_{\text{ft}} - f_{\text{s}})}_{\text{DA objective}} + \gamma_{\text{DB}} \cdot \underbrace{(f_{\text{db}} - f_{\text{ft}})}_{\text{DB objective}}$ ▷ Eq. 9
  $y_t \leftarrow$ Sample from $\text{softmax}(f_t)$
  $t \leftarrow t + 1$
return y

Model size	HQA	MARCO	NQ
1B	39.66	25.12	32.12±5.53
8B	70.94	46.31	70.20±1.34
+ MS	73.89	48.77	67.69±2.45
% Gain	4.2%	5.3%	-3.6%
405B	**84.98**	**52.46**	**76.45** ±3.31
% Gain	19.8%	13.3%	8.9%

Table 1: Impact of model scaling: In the LLaMA 3 model family, we compare the smallest model 1B and the largest model 405B with our main target model, 8B.

Activation			Benchmark		
MS	DA	DB	HQA	MARCO	NQ
			70.94	46.31	70.20±1.34
		✓	70.44	49.75	**70.59±0.41**
	✓		78.08	**50.49**	66.55±0.64
	✓	✓	79.56	49.51	68.67±0.56
✓			73.89	48.77	67.69±2.45
✓	✓	✓	**81.03**	49.75	64.98±0.75

Table 2: Task-level adaptation of GRAD on different benchmarks, with LLaMA 3.1 8B as the base LLM. The combination of objectives determined by GRAD's task-wise adaptation is underlined, while the best results are **boldfaced**.

provide more details in Appendix A.

4.2 Experimental Results

In this section, we aim to answer the following research questions:

- (RQ1) Do decoding objectives implemented with smaller models align with RAG strategies?
- (RQ2) Does GRAD realize effective task-adaptive combination of objectives?
- (RQ3) Does GRAD generalize to private, domain-specific RAG scenarios?
- (RQ4) Does GRAD achieve a better cost-performance trade-off?

4.2.1 Alignment of Decoding and RAG Objectives

Before evaluating the combined effect of RAG objectives, we first verify whether each token-level decoding objective in GRAD aligns with its corresponding RAG goal, MS, DA, or DB. Each individual objective here can serve as a baseline for comparison against GRAD.

MS Table 1 shows that larger models consistently improve accuracy and CD aligns with MS, though its effectiveness varies by task. While CD enhances HQA and MARCO (by 4.2% and 5.3%), it hurts accuracy on NQ, likely due to *diminishing returns* from actual scaling. NQ exhibits a small accuracy gap (8.9%) between 8B and 405B models, whereas HQA's is much larger (19.8%). CD's diminishing returns align with those of scaling.

DA Next, Table 2 shows that domain-specific finetuning is helpful on HQA and MARCO, where well-aligned training data $\mathcal{D}$ is provided in the dataset. Not surprisingly, DA is not effective for NQ and PQ with no such data, an out-of-

domain challenge which is further discussed in Section 4.2.3.

DB Our CCD objective effectively reduces position bias as intended, which is measured by the standard deviation across varying gold context positions on NQ. Table 2 shows that activating DB significantly decreases this variance, suggesting that CCD mitigates position bias.

These results show that steering models no longer requires large, hand-crafted datasets: Pairwise contrast between models replaces the role of data to isolate and optimize an objective directly through decoding.

Qualitative examples We also provide case studies in Appendix F, to examine how these objectives translate to real impact on the targeted biases, beyond mere accuracy: Figure 2 (domain adaptation on HQA) and 3 (reduced position sensitivity under DB when the gold passage moves).

4.2.2 Adaptive Optimization with GRAD

GRAD dynamically adjusts to different scenarios by combining objectives adaptively, as shown in Table 2, where applying all objectives is not universally optimal. On HQA and MARCO, where well-aligned training data is available, DA is most beneficial. HQA performs best with all three objectives (MS, DA, DB), while MARCO sees no added gains beyond DA. On NQ, DA proves counterproductive, whereas DB improves performance by mitigating position bias.[3] We further show natural extensibility of GRAD by adding a fourth objective, FA (format adherence); as shown in Appendix H, GRAD integrates the new objective without changes, and adaptively optimizes to find the desirable combination.

The adaptive mechanism of GRAD effectively prevents over-steering of the base model's prediction as noted in Section 3.4: Manual inspection of 100 random samples on NQ contained no degenerate outputs, which often indicate strong conflicts (see Appendix C).

More analyses on the adaptation mechanism of GRAD are provided in Appendix D, where we (1) compare GRAD with other strategies for combining several model predictions, such as ensembling, (2) show the stability of adaptation across different random seeds, and (3) show the effect of choosing different ordering of objectives in adaptation.

[3]See Appendix C for full position-wise results.

Activation			PQ	
MS	DA	DB	Synthesized $\mathcal{D}$	Transferred $\mathcal{D}$
			60.83	60.83
		✓	63.33	**61.67**
	✓		64.17	51.67
	✓	✓	64.17	53.33
✓			61.67	61.67
✓	✓	✓	**65.00**	52.50

Table 3: Performance of GRAD on PQ, with synthesized (from PDF document collection) or transferred (from HQA) training set $\mathcal{D}$.

	HQA	
Model	Latency (s)	Acc
8B	0.81	70.94
GRAD (8B+1B)	1.16	81.03
70B	5.75	84.24

Table 4: End-to-end latency of GRAD with 8B+1B, compared to 8B and 70B models.

4.2.3 Private-Data RAG Scenarios

Using the PQ dataset, we analyze how GRAD and its components react to a private, domain-specific RAG scenario where the LLM has to adapt to unseen data.

Table 3 shows accuracy of GRAD with different set of components activated on PQ, where the SLMs are either finetuned on a synthesized (left) or transferred (right, reusing HQA) $\mathcal{D}$.

Table 2 and 3 first reveal that, both transferred (HQA) and synthesized (PQ) $\mathcal{D}$ can enhance RAG quality in private-data scenarios. In addition, the DB objective generalizes better across domains than the DA objective, as seen with NQ in Section 4.2.2: Table 3 also shows that while SLMs finetuned on transferred $\mathcal{D}$ from HQA degrade performance under DA, they still yield gains with DB, though less than with synthesized data. We also observe consistent results on repurposed HQA, where we intentionally use SLMs trained on MARCO to replicate the private setting: Those results are available in Table 15, in Appendix G. Our results suggest that, if training dataset $\mathcal{D}$ well-aligned to test-data distribution is not available, synthesizing closely aligned data can be a viable alternative to overcome the unavailability of training data.

4.2.4 Latency and Cost

Finally, we present end-to-end latency of GRAD in Table 4, measured under our current sequential implementation, and thus providing a conservative

Model	HQA	MARCO	NQ	PQ
θ^s	39.66	25.12	32.12	25.83
θ^s_{ft}	78.82	28.57	47.39	35.00
θ^s_{cord}	78.82	35.22	49.55	36.67
θ (8B)	70.94	46.31	70.20	60.83
GRAD	**81.03**	**50.49**	**70.59**	**65.00**

Table 5: Performance of finetuned 1B models: While they cannot outperform larger 8B in most cases (underlined), they can still guide 8B to generate improved outputs (**boldfaced**). For comparison, accuracy of GRAD is also presented in the last row.

SLM θ^s	HQA Acc
`LLaMA-3.2-1B`	81.03
`LLaMA-3.2-3B`	82.02

Table 6: Accuracy of GRAD with 8B paired with SLMs of different sizes.

Activation			Benchmark			
MS	DA	DB	HQA	MARCO	NQ	PQ
70B			82.76	47.04	72.17	73.33
		✓	83.25	50.74	**73.59**	74.17
	✓		**85.22**	**51.72**	69.26	**75.00**
	✓	✓	**85.22**	50.49	69.93	**75.00**
✓			81.77	48.28	71.67	72.50
✓	✓	✓	83.25	49.26	70.94	**75.00**
405B			84.98	52.46	76.45	74.17

Table 7: Accuracy of GRAD with LLaMA 3.1 70B used as the base LLM. For comparison, the accuracy of 405B model is presented on the last line.

estimate of its inference-time overhead. Even in this setting, GRAD incurs small inference-time overhead compared to base LLM but is much more effective than a larger model. GRAD with 8B + 1B retains approximately 76% of the performance gains of a 70B model, while achieving 5x lower latency. Moreover, the relative overhead of using a fixed SLM (1B) diminishes with larger base models—for instance, adding three 1B models to an 8B base introduces about 40% additional parameters, compared to only 5% with a 70B base.

Cost per objective Each objective contributes a similar per-token cost: enabling an objective adds one small-model invocation and a lightweight weighted-logit combine. Since our current implementation runs the auxiliary models sequentially, the added latency over the 8B base therefore comes almost entirely from three additional forward passes, one per objective (MS, DA, DB); these passes are independent given the same prefix, so the wall-clock overhead can be further reduced in a parallel implementation.

5 Analysis

Beyond the main results, we examine the mechanism and generality of GRAD, showing that its gains arise from transferable guidance signals rather than from a narrow choice of helper model, backbone family, or deployment setting.

Guidance from weaker models Table 5 indeed shows that while finetuned SLMs show improvements over their base model, they still struggle to outperform the larger base LLM (8B) on most tasks. However, by leveraging their combined knowledge, seemingly weak signals from these smaller models can improve much stronger models, reaffirming the effectiveness of GRAD, capturing the most significant adaptation signals in logit space. Finally, each component in GRAD may be upgraded to capture even richer signals at increased costs.

Choice of SLM/LLM Regarding SLM, Table 6 contradicts CD, where the larger scale gap between the LLM and SLM reportedly resulted in the best outcomes (Li et al., 2023; O'Brien and Lewis, 2023). Employing a more capable model as the SLM pays off for GRAD.

Table 7 demonstrates that GRAD remains effective when the base LLM θ is 70B, not just when θ is 8B, consistently improving performance across different LLM choices. It yields comparable results compared to actually scaling the model size: Combination of 8B and 1B matches the performance of 70B, and combination of 70B and 1B matches 405B. For instance, GRAD utilizing 8B and 1B achieves 81.03 on HQA (Table 5), which is comparable to 70B's 82.76, closing the large gap between 8B (70.94) and 70B. Similarly, GRAD with 70B and 1B achieves 85.22, slightly outperforming 84.98 of 405B on HQA. These results demonstrate that GRAD effectively balances performance and efficiency.

Generalization across model families Finally, we show that benefits of GRAD are orthogonal to the choice of backbone model family. Table 8 shows GRAD combined with Qwen-2.5 models, successfully improving the 7B model's performance over that of 14B/32B models, with 1.5B models. The results confirm GRAD general-

Activation			Benchmark		
MS	DA	DB	HQA	MARCO	NQ
7B			67.73	48.77	67.93±2.57
		✓	67.24	50.74	**69.86**±0.54
	✓		**76.60**	**51.72**	64.57±2.43
	✓	✓	**76.60**	51.23	64.88±1.67
14B			74.38	49.01	71.48±1.35
32B			75.62	50.25	76.11±1.70

Table 8: GRAD is also effective when Qwen 2.5 7B is the base LLM and 1.5B is used for SLM.

izes across model families, other than the LLaMA model family mainly considered, exhibiting consistent trends.

GRAD under restricted logit access GRAD, like other decoding-time techniques involving auxiliary models, assumes access to logits. Here, we discuss how GRAD can be extended when logit access is restricted, including both partial log-probability access and fully blind settings.

Many APIs return top-k log-probabilities $\log p_t(y)$ at each decoding step. By definition of the softmax,

$$\log p_t(y) = f_t(y) - \log Z_t, \tag{15}$$

where $f_t(y)$ denotes the unnormalized logit and $Z_t = \sum_{y' \in V} \exp(f_t(y'))$. Thus, $f_t(y) = \log p_t(y) + \log Z_t$ up to an additive constant. In GRAD's guided distribution $\mathrm{softmax}(f_t + \sum_i \gamma_i R_{i,t})$, this constant cancels out, allowing us to replace inaccessible logits $f_t(y)$ with available log-probabilities $\log p_t(y)$.

At decoding step t, let K denote the set of tokens for which the base LLM provides log-probabilities. Optionally, we expand K to the union of top-k tokens from the auxiliary models. For each $y \in K$, we compute reward terms $R_{i,t}(y)$ using the available log-probabilities or logits from the corresponding models. If a model does not provide a score for token y, we assign it a floor value (e.g., $-\infty$). We then form sparse guided scores

$$\tilde{f}_t(y) = \log p_t(y) + \sum_i \gamma_i R_{i,t}(y), \qquad y \in K, \tag{16}$$

apply $\mathrm{softmax}$ over K, and decode (e.g., greedily or with beam search) from this sparse distribution.

This procedure approximates full-vocabulary decoding by ignoring tokens outside K. Since GRAD is typically used with greedy decoding, where only the highest-scoring token matters, this sparse approximation is often sufficient in practice.

When neither logits nor top-k log-probabilities are available, a practical alternative is to train a surrogate model to imitate the black-box target model, and then apply GRAD using the surrogate's accessible logits. Recent work suggests that such black-box distillation can be effective even without logit-level supervision, either by learning to match teacher and student text distributions adversarially (Ye et al., 2025) or by fine-tuning a surrogate on a small set of collected black-box outputs to reduce surrogate-target mismatch (Zeng et al., 2024). In this case, the surrogate provides the token-level interface required by GRAD, extending the framework in principle to fully blind settings.

GRAD with LoRA adapters We also consider realizing each objective as a LoRA adapter attached to the same base LLM. This avoids cross-model tokenizer mismatch and removes the need for separate auxiliary models. Appendix H compares logit-space composition with parameter-space merging of adapters and shows that this is a promising direction.

6 Conclusion

We studied a unified decoding-time framework that integrates diverse RAG strategies of model scaling, domain adaptation, and positional debiasing as adaptive decoding objectives, based on smaller models. This translation eliminates costly training and inference overheads while enabling dynamic selection of objectives based on their contribution to the task. Our approach introduces a new Pareto frontier, by balancing efficiency and performance. Our method was validated in RAG benchmarks as well as a real-life scenario with unseen domain and no training data.

Limitations

First, our method assumes access to token logits. While we described a practical adaptation for GRAD in Section 5, that replaces full-vocabulary operations with softmax over observed tokens and discusses surrogate options, empirically validating those settings remains as future effort.

Also, a related question is whether GRAD can operate if the base and helper models have different vocabulary/tokenizers. Extending decoding-time

fusion to mismatched vocabularies is an emerging direction; recent work has explored token-level collaboration across models with different vocabularies (Bian et al., 2025). Such approaches suggest that an alignment layer, e.g., based on prefix grouping or token mapping, could enable auxiliary models with different tokenizers to provide compatible guidance within GRAD. We leave this extension to future work.

Finally, our main experiments focus on single-turn RAG. We provide initial iterative/multi-hop results using HQA in Appendix E to show that single-turn gains carry over, but fully generalizing the adaptation mechanism to multi-turn pipelines remains open.

References

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2018. Ms marco: A human generated machine reading comprehension dataset. *Preprint*, arXiv:1611.09268.

Yuang Bian, Yupian Lin, Jingping Liu, and Tong Ruan. 2025. PToco: Prefix-based token-level collaboration enhances reasoning for multi-LLMs. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 8326–8335, Abu Dhabi, UAE. Association for Computational Linguistics.

Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas Joglekar, Jan Leike, Ilya Sutskever, and Jeffrey Wu. 2024. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 4971–5012. PMLR.

Haw-Shiuan Chang, Nanyun Peng, Mohit Bansal, Anil Ramakrishna, and Tagyoung Chung. 2024. Explaining and improving contrastive decoding by extrapolating the probabilities of a huge and hypothetical LM. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8503–8526, Miami, Florida, USA. Association for Computational Linguistics.

Qinyuan Cheng, Xiaogui Yang, Tianxiang Sun, Linyang Li, and Xipeng Qiu. 2023. Improving contrastive learning of sentence embeddings from AI feedback. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11122–11138, Toronto, Canada. Association for Computational Linguistics.

Chenghao Fan, Zhenyi Lu, Wei Wei, Jie Tian, Xiaoye Qu, Dangyang Chen, and Yu Cheng. 2024. On giant's shoulders: Effortless weak to strong by dynamic logits fusion. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Cheng-Yu Hsieh, Yung-Sung Chuang, Chun-Liang Li, Zifeng Wang, Long Le, Abhishek Kumar, James Glass, Alexander Ratner, Chen-Yu Lee, Ranjay Krishna, and Tomas Pfister. 2024. Found in the middle: Calibrating positional attention bias improves long context utilization. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14982–14995, Bangkok, Thailand. Association for Computational Linguistics.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Maxim Khanov, Jirayu Burapacheep, and Yixuan Li. 2024. ARGS: alignment as reward-guided search. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Taehyeon Kim, Joonkee Kim, Gihun Lee, and Se-Young Yun. 2024. Instructive decoding: Instruction-tuned large language models are self-refiner from noisy instructions. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Youngwon Lee, Seung-won Hwang, Daniel F Campos, Filip Graliński, Zhewei Yao, and Yuxiong He. 2025a. CORD: Balancing COnsistency and rank distillation for robust retrieval-augmented generation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 787–796, Albuquerque, New Mexico. Association for Computational Linguistics.

Youngwon Lee, Seung-won Hwang, Daniel F Campos, Filip Graliński, Zhewei Yao, and Yuxiong He. 2025b. Inference scaling for bridging retrieval and augmented generation. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 7324–7339, Albuquerque, New Mexico. Association for Computational Linguistics.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. 2023. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 19274–19286. PMLR.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2023. Contrastive decoding: Open-ended text generation as optimization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12286–12312, Toronto, Canada. Association for Computational Linguistics.

Alisa Liu, Xiaochuang Han, Yizhong Wang, Yulia Tsvetkov, Yejin Choi, and Noah A. Smith. 2024a. Tuning language models by proxy. In *First Conference on Language Modeling*.

Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A. Smith, and Yejin Choi. 2021. DExperts: Decoding-time controlled text generation with experts and anti-experts. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6691–6706, Online. Association for Computational Linguistics.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024b. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Kelong Mao, Zheng Liu, Hongjin Qian, Fengran Mo, Chenlong Deng, and Zhicheng Dou. 2024. RAG-studio: Towards in-domain adaptation of retrieval augmented generation through self-alignment. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 725–735, Miami, Florida, USA. Association for Computational Linguistics.

Sean O'Brien and Mike Lewis. 2023. Contrastive decoding improves reasoning in large language models. *CoRR*, abs/2309.09117.

Ahmad Rashid, Ruotian Wu, Julia Grosse, Hongliang Li, Agustinus Kristiadi, and Pascal Poupart. 2025. A critical look at tokenwise reward-guided text generation.

Ruizhe Shi, Yifang Chen, Yushi Hu, Alisa Liu, Hanna Hajishirzi, Noah A. Smith, and Simon S. Du. 2024. Decoding-time language model alignment with multiple objectives. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Shamane Siriwardhana, Rivindu Weerasekera, Elliott Wen, Tharindu Kaluarachchi, Rajib Rana, and Suranga Nanayakkara. 2023. Improving the domain adaptation of retrieval augmented generation (RAG) models for open domain question answering. *Transactions of the Association for Computational Linguistics*, 11:1–17.

Seamus Somerstep, Felipe Maia Polo, Moulinath Banerjee, Yaacov Ritov, Mikhail Yurochkin, and Yuekai Sun. 2025. A transfer learning framework for weak to strong generalization. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Yuancheng Xu, Udari Madhushani Sehwag, Alec Koppel, Sicheng Zhu, Bang An, Furong Huang, and Sumitra Ganesh. 2025. GenARM: Reward guided generation with autoregressive reward model for test-time alignment. In *The Thirteenth International Conference on Learning Representations*.

Xiao Yang, Kai Sun, Hao Xin, Yushi Sun, Nikita Bhalla, Xiangsen Chen, Sajal Choudhary, Rongze Daniel Gui, Ziran Will Jiang, Ziyu Jiang, Lingkun Kong, Brian Moran, Jiaqi Wang, Yifan Ethan Xu, An Yan, Chenyu Yang, Eting Yuan, Hanwen Zha, Nan Tang, Lei Chen, Nicolas Scheffer, Yue Liu, Nirav Shah, Rakesh Wanga, Anuj Kumar, Wen-tau Yih, and Xin Luna Dong. 2024. CRAG - comprehensive RAG benchmark. *CoRR*, abs/2406.04744.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Tianzhu Ye, Li Dong, Zewen Chi, Xun Wu, Shaohan Huang, and Furu Wei. 2025. Black-box on-policy distillation of large language models. *CoRR*, abs/2511.10643.

Cong Zeng, Shengkun Tang, Xianjun Yang, Yuanzhou Chen, Yiyou Sun, Zhiqiang Xu, Yao Li, Haifeng Chen, Wei Cheng, and Dongkuan Xu. 2024. DALD: improving logits-based detector without logits from black-box llms. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Tianjun Zhang, Shishir G. Patil, Naman Jain, Sheng Shen, Matei Zaharia, Ion Stoica, and Joseph E. Gonzalez. 2024. Raft: Adapting language model to domain specific rag. *Preprint*, arXiv:2403.10131.

Zhanhui Zhou, Zhixuan Liu, Jie Liu, Zhichen Dong, Chao Yang, and Yu Qiao. 2024. Weak-to-strong search: Align large language models via searching over small language models. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Algorithm 2 Task-Level Activation of Decoding Objectives

Require: Small sample $\mathcal{S}$ (default $|\mathcal{S}|$=100), objective order $\mathcal{O}$=[DA $\rightarrow$ DB $\rightarrow$ MS], candidate set Γ (default $\{0.1, 0.2, 0.3, 0.5, 1, 1.5, 2\}$), distance threshold τ (default 0.1), activation threshold ρ (default 0.5), optional dev labels $\mathcal{L}$

Ensure: Activated set $\mathcal{A}$, effective weights $\gamma_i = \gamma_i^0 \gamma_i^1$ (magnitudes $\gamma_i^0 \in \Gamma$, switches $\gamma_i^1 \in \{0, 1\}$)

$\mathcal{A} \leftarrow \emptyset$
for each $i \in \mathcal{O}$ **in order do**
  if dev labels $\mathcal{L}$ are available **then** ▷ Evaluate i *conditioned* on previously activated $\mathcal{A}$
    $\gamma_i^0 \leftarrow \arg\max_{\gamma \in \Gamma}\ \mathrm{Acc}(\mathcal{A} \cup \{i\}; \gamma)$
    $\mathrm{Acc}_i^{\mathrm{on}} \leftarrow \mathrm{Acc}(\mathcal{A} \cup \{i\}; \gamma_i^0)$
    $\mathrm{Acc}_i^{\mathrm{off}} \leftarrow \mathrm{Acc}(\mathcal{A})$
    if $\mathrm{Acc}_i^{\mathrm{on}} \geq \mathrm{Acc}_i^{\mathrm{off}}$ **then**
      $\mathcal{A} \leftarrow \mathcal{A} \cup \{i\}$; set $\gamma_i^1 \leftarrow 1$
    else
      $\gamma_i^1 \leftarrow 0$
  else
    for each $\gamma \in \Gamma$ **do** ▷ Label-free: select magnitude by divergence alignment, then decide activation
      for each $x \in \mathcal{S}$ **do**
        $y \leftarrow \mathrm{GreedyDecode}(x;\ \text{objectives} = \mathcal{A} \cup \{i\},\ \gamma_i = \gamma,\ \{\gamma_j\}_{j \in \mathcal{A}}\ \text{fixed})$
        for $t = 1 \ldots T(y)$ **do**
          compute $d_{i,\gamma}^{(t)}$ from Eq. 12 conditioned on $(x, y_{<t})$
        $p_{i,\gamma}(x) \leftarrow \frac{1}{T(y)} \sum_{t=1}^{T(y)} \mathbb{I}[d_{i,\gamma}^{(t)} < \tau]$
      $\bar{p}_{i,\gamma} \leftarrow \frac{1}{|\mathcal{S}|} \sum_{x \in \mathcal{S}} p_{i,\gamma}(x)$
    $\gamma_i^0 \leftarrow \arg\max_{\gamma \in \Gamma}\ \bar{p}_{i,\gamma}$
    if $\max_{\gamma \in \Gamma} \bar{p}_{i,\gamma} \geq \rho$ **then**
      $\mathcal{A} \leftarrow \mathcal{A} \cup \{i\}$; set $\gamma_i^1 \leftarrow 1$
    else
      set $\gamma_i^1 \leftarrow 0$
  $\gamma_i = \gamma_i^0 \gamma_i^1$
return $\mathcal{A}$, $\{\gamma_i\}$

A Implementation Details

Inputs Across all benchmarks, the LLM receives the top-10 retrieved contexts. For NQ, following prior work (Liu et al., 2024b), we vary the position of the gold context (0, 2, 4, 6, and 9) to assess the model's ability to utilize the full input contexts. We report average performance across these positions along with standard deviation, as a measure of its sensitivity to input ordering.

Models We use the publicly available LLaMA-3 model family as the backbone generator,[4] employing greedy decoding for zero-shot response generation to ensure reproducibility, and also study model generalization on Qwen family. Following prior works (Yang et al., 2024; Lee et al., 2025a), we evaluate performance using accuracy scores obtained via LLM-as-a-judge.

SLM finetuning For the DA objective, we finetune SLMs on the respective training sets of each benchmark. Following Lee et al. (2025a), we use LoRA (Hu et al., 2022) with rank $r = 8$ and $\alpha = 32$, dropout rate of 0.1, training for 5 epochs with learning rate of 1e-4 and effective batch size of 4, with weight decay of 0.01 applied. In addition, for the DB objective, the coefficient λ for controlling the contribution of consistency loss was set to 10, with the noise degree for interpolating contexts of 0.5.

[4] Specifically, we consider 3.2 1B, 3B, 3.1 8B, 70B, and 405B Instruct models.

Activation			Index of Gold Context					
MS	DA	DB	0	2	4	6	9	Avg
			71.18	70.69	71.43	69.46	68.23	70.20±1.34
		✓	70.69	70.20	70.20	71.18	70.69	**70.59±0.41**
	✓		66.16	66.65	67.14	67.14	65.67	66.55±0.64
	✓	✓	68.72	68.97	67.73	68.72	69.21	68.67±0.56
✓			70.69	67.98	68.72	67.00	64.04	67.69±2.45
✓	✓	✓	64.53	64.53	64.29	65.52	66.01	64.98±0.75

Table 9: Full position-wise results on NQ.

PQ details The retrieval corpus is obtained from 215 internal IT PDF documents, where each PDF document is sourced from HTML documents, resulting in 1,857 passages after parsing to raw text and chunking. Each passage is truncated at max of 1,000 tokens with 100 token overlap between passages. The ground-truth answers have been labeled by human annotators, ensuring there exists a gold passage that can answer the question for each query.

B More on Contrastive Decoding

The original formulation proposed by Li et al. (2023) defined CD by (1) combining signals in log-probability space and (2) subtracting the amateur model's log-probabilities without weighting. Subsequent works (O'Brien and Lewis, 2023; Chang et al., 2024) extended this by introducing temperature, equivalent to the reciprocal of a weight parameter γ in our formulation, and by combining signals in logit space, aligning with related approaches. This setting can be expressed as:

$$p(\cdot \mid y_{<t}) = \text{softmax}(f_t - \gamma f_t^-), \qquad (17)$$

which effectively removes the γf_t^+ term from GRAD's scaling component. We unify these into a weighted logit combination, which has additional benefit of generalizing to arbitrary weighted unification of multiple objectives.

$$p(\cdot \mid y_{<t}) = \text{softmax}(\gamma_1 f_{1,t} + \gamma_2 f_{2,t}), \quad p \propto p_1^{\gamma_1} p_2^{\gamma_2}, \qquad (18)$$

where p_1 and p_2 are probability distributions derived from f_1 and f_2, respectively. This equivalence highlights why constraining the sum of weights to 1 ensures a stable distribution; any deviation from this introduces unintended temperature effects. In particular, if the sum equals zero as in the original unweighted subtraction, the resulting distribution can become highly unstable, especially for closed-form generation tasks. To mitigate this, Li et al. (2023) introduced corrective measures such as filtering out tokens deemed improbable by the expert (larger) model.

Model	NQ (20)	PQ
8B	68.08±1.60	60.28±1.04
8B + DB	68.87±0.45	61.94±0.29

Table 10: Effectiveness of CCD on distractor-heavier settings, NQ with 20 passages and PQ.

C Detailed Analyses of NQ

First, we provide the exhaustive results on NQ benchmark, showing the accuracy per different relative position of the gold context in Table 9. To obtain these results, we hold retrieval fixed and inject the gold passage among contexts while shifting only its position. Here, the lower standard deviation indicates more robust, and thus more *faithful*, use of the true evidence. In other words, this naturally provides a measurement of the model's faithfulness.

Next, we describe in more detail the manual check conducted on 100 generated samples on NQ, generated with GRAD (where only the DB objective is activated). NQ has been chosen to separately observe the impact of the DB objective, which is most likely to deviate from the base model's prediction and interfere with it, to produce unreliable or degenerate distribution. Specifically, we have looked for repetition of a token sequence in the output, which is the most widely reported as a degenerate behavior of a language model. No samples were flagged as degenerate, which aligns with the activation guard in Section 3.4, which suppresses an objective when the divergence signal indicates misalignment.

Finally, in order to further stress-test robustness of DB objective and CCD, we also evaluate on longer inputs with 20 passages (i.e., 1 gold pas-

Ensembling with θ			HQA
$+\theta^s_{\text{db}}$	$+\theta^s_{\text{ft}}$	$+\theta^s$	Acc
θ only			70.94
Model Ensembling			
✓	✓	✓	<u>69.21</u>
✓	✓		73.65
✓			79.31
	✓		74.14
GRAD			**81.03**

Table 11: Model ensemble (1) fails to capture the negative effect of θ^s (<u>underlined</u>), and (2) positive components perform better when used separately (shaded).

Benchmark	DA	$\rightarrow$ DB	$\rightarrow$ MS
HQA	84.3 (2.9)	77.3 (2.5)	67.0 (3.6)
MARCO	81.7 (2.1)	23.7 (4.7)	28.3 (4.5)
NQ	25.7 (5.0)	66.7 (6.1)	20.3 (3.8)

Table 12: Stability of adaptation with respect to random sample choice: The numbers show the average proportion of examples (%) with high activation, along with standard deviation across three runs.

sage and 19 distractors) averaged across different gold positions (0/4/9/14/19), and also on PQ by randomly shuffling the input passages (averaging 3 runs). As shown in Table 10, CCD continues to mitigate position sensitivity while also improving mean accuracy under these even more distractor-heavy settings.

D More Analyses on Adaptive Optimization

GRAD vs. ensembling The effective adaptation achieved by GRAD can also be compared with model ensembling, in which the outputs from multiple models are simply averaged. Table 11 shows that ensembling finetuned models does not yield meaningful synergy (shaded rows), and incorporating logits from the base small model (θ^s) degrades performance (underlined). In contrast, GRAD leverages θ^s as a contrasting reference, producing positive gains and consistently outperforming model ensembling.

Stability of adaptation Here, we demonstrate that the KL-based adaptation mechanism of GRAD is stable, as the same combination of objectives is consistently returned across different random choices of examples. Table 12 reports the average and standard deviation of the proportion of examples with high activation, using 100 randomly selected examples across three runs.

Ordering	Activated	Accuracy
DA $\rightarrow$ DB $\rightarrow$ MS (Ours)	{DA}	**50.49**
DA $\rightarrow$ MS $\rightarrow$ DB	{DA}	**50.49**
DB $\rightarrow$ DA $\rightarrow$ MS	{DB}	49.75
DB $\rightarrow$ MS $\rightarrow$ DA	{DB}	49.75
MS $\rightarrow$ DA $\rightarrow$ DB	{MS}	48.77
MS $\rightarrow$ DB $\rightarrow$ DA	{MS}	48.77

Table 13: Accuracy under different objective ordering strategies and the corresponding activation on MARCO.

Activation	1st hop	2nd hop	All
None	86%	85%	73%
MS + DA + DB	88%	88%	77%

Table 14: Accuracy of GRAD (all objectives activated) with iterative generation on HQA.

Effect of objective ordering As shown in Algorithm 1 and 2, GRAD considers the objectives in the order of DA, DB, followed by MS; here, we validate this design choice through an ablation study. Table 13 shows the activations resulting from different orders of objectives and their corresponding task performance. The results are from MARCO, only changing the order in which the objectives are considered. This ablation supports our choice of ordering.

E Results on Iterative RAG

While our focus on single-turn setting was to align and compare with prior work on contrastive objectives and position bias mitigation, we also present results in an iterative RAG task, based on HQA with decomposition and iterations, with the same 8B and 1B LLaMA models used in the main experiments. We followed the setting of works on iterative RAG mostly evaluated on multi-hop QA benchmarks; we employed GPT-4o to decompose the 2-hop questions in HQA, and identified those with sequential dependencies, i.e., queries that subquestion 2 can be answered only after correctly answering subquestion 1. Questions without sequential dependency between the subquestions were not considered. The subquestions were provided to the LLaMA model for generation.

Single-turn accuracy on the "1st hop" column in Table 14 is obtained over 100 examples, and accuracy on the "2nd hop" column is obtained over questions where the first subquestion was successfully answered. The combined multi-turn accuracy for the original 2-hop query is presented in the "All" column. This result supports that improvement in

Activation			HQA
MS	DA	DB	Acc
			70.94
	✓		66.01
		✓	72.66
	✓	✓	66.50
✓			73.89
✓		✓	**74.88**

Table 15: Results on repurposed HQA: SLMs are trained on MARCO instead of HQA.

Activation				HQA
MS	DA	DB	FA	Acc w/ formatting
				69.21
			✓	74.14
	✓		✓	76.85
	✓	✓	✓	77.34
✓	✓	✓	✓	**77.83**

Table 16: Results on HQA, with formatting requirements.

Method	Acc on HQA
Orig GRAD	**79.56**
GRAD w/ adapters	79.06
Merged adapters	76.85

Table 17: Adapter-based objectives for GRAD. Replacing helper models with adapters on a shared base LLM enables replacing decoding-time composition with parameter-space merging.

single-turn generation by GRAD does translate to improvement in multi-turn generation.

F Qualitative Analysis

In this section, we provide some qualitative examples showcasing the benefits of GRAD.

Regarding the effect of DA objective, Figure 2 qualitatively shows how it shifts model prediction towards data/task-specific expectations: HQA is a multi-hop QA dataset, where each query can be decomposed into two subquestions. As shown in the above example, the base model often stops after answering the first-hop question; this behavior is effectively suppressed with the guidance of a finetuned SLM.

Regarding debias through DB objective, Figure 3 qualitatively shows how debias objective improves proportionate utilization of contexts. The base 8B model correctly replies with "October 2012" when the gold passage is placed at index 0; for other positions 2, 4, 6 and 9, its answer was "March 2008," an incorrect answer possibly derived from a shortcut in passage 2. Though passage 2 is an irrelevant context, it has been ranked higher by the retrieval system, and the base model's prediction has been highly influenced by the beginning of the context. In contrast, 8B with DB objective activated, consistently generated the correct answer "October 2012" regardless of the position of the gold passage.

G Repurposing HQA for Private Setting

To provide further evidence of the effectiveness of GRAD in a private (no in-domain training data) setting, we repurpose HQA by training the SLMs on MARCO (out-of-domain) and evaluating on HQA. This mirrors realistic deployments where high-quality, task-matched labels are unavailable and only related-domain supervision exists.

Table 15 shows that GRAD isolates helpful signals under such setting as well: DB and MS remain beneficial, while DA is suppressed, yielding consistent improvements over the base model.

H Extensibility of GRAD

Here, we demonstrate that GRAD can be extended both by incorporating new objectives and by changing how objectives are realized.

Adding a new objective (FA) We consider a simple *format adherence* (FA) objective that nudges generation toward well-formed, sentence-level outputs. FA is added alongside existing objectives (MS, DA, DB) without any change to GRAD; activation still follows the same task-level adaptation.

Table 16 reports accuracy under the formatting constraint, where an LLM judge is prompted to evaluate the formatting as well, and an answer is considered correct if it satisfies both the correctness and the formatting requirements. By considering the new FA objective first, GRAD easily determines the most favorable combination of activating all.

GRAD with LoRA adapters We also consider an alternative realization of objectives, where each objective is implemented as a LoRA adapter attached to the same base LLM. This avoids cross-model tokenizer mismatch and removes the need for separate auxiliary models. In this setting, objectives can be combined either (1) at decoding time

Qualitative Analysis on HQA for DA Objective

Question: Roger O. Egeberg was Assistant Secretary for Health and Scientific Affairs during the administration of a president that served during what years?
Ground-truth answer: “1969 until 1974”
Answer from base 8B: “Nixon”
Answer from 8B + FT: “1969 until 1974, when he resigned from office, the only U.S. president to do so.”

Figure 2: An example from HQA, showing the effect of DA objective.

Qualitative Analysis on NQ for DB Objective

Question: when did the first wireless beats come out?
Ground-truth answer: October 2012
(Gold Passage, “Beats Electronics”) In October 2012, Beats unveiled its first two self-developed products, "Beats Executive" headphones and "Beats Pill" wireless speakers—Iovine believed that ...
(Passage 1, “Beats Electronics”) ... The appointment of a new chief operating officer (COO), a role previously filled by Wood, was announced in early November 2013. ...
(Passage 2, “The Pacemaker”) It was also at the Amsterdam dance event. Originally scheduled for release in February 2008, slight delays pushed it back to March 2008 when the first units were shipped. ...
...
(Passage 9, “Interscope Records”) 2006, Dre and Iovine established Beats Electronics. Dre had been approached by his attorney to start a line of sneakers, and when he told Iovine about the idea, ...

Figure 3: An example from NQ, showing the effect of DB objective.

via GRAD-style logit-space composition, or (2) in parameter space by merging multiple adapters into a single set of weights, eliminating additional decoding-time overhead.

Table 17 compares these alternatives on HQA, using DA and DB objectives where both are enabled. Overall, these results suggest that LoRA adapters provide a promising lightweight realization of objectives in GRAD (second row). Although merging them in parameter space slightly hurts performance compared to decoding-time composition (third row), it still offers substantial improvements while avoiding the decoding-time overhead.