EAGPO: EDIT-ANCHORED GENERALIZATION POLICY OPTIMIZATION FOR KNOWLEDGE EDITING

Anonymous authors
Paper under double-blind review

ABSTRACT

Knowledge editing aims to precisely update factual knowledge in large language models. However, existing methods struggle to generalize the editing knowledge to semantic variants, limiting their practical application. This fragility often arises because these methods optimize solely for a single target expression, neglecting the broader semantic neighborhood that real-world queries inhabit. To bridge this gap, we propose EAGPO, a novel framework that enhances knowledge editing generalization to semantic variants through **E**dit-**A**nchored **G**eneralization **P**olicy **O**ptimization. It first employs a standard knowledge editing method to precisely integrate the new knowledge. Then, it performs reinforcement learning training via two core components: an Anchor-Generalization Training (AGT) mechanism that optimizes over structured query clusters, and a Policy Gradient Alignment Advantage (PGAA) module that stabilizes updates through geometric gradient alignment. Theoretically, EAGPO strengthens knowledge protection, provides a tighter generalization error bound, and reduces policy gradient variance. Empirical results on the KnowEdit benchmark show that EAGPO significantly outperforms strong baselines across three types generalization tasks, improving generalization performance by 25%–35% over the original knowledge editing methods, while achieving a 25% improvement in computational efficiency over GRPO training. Our code is available at EAGPO [1].

1 INTRODUCTION

Large language models (LLMs) are typically trained on static datasets, which causes their internal knowledge to become fixed upon completion. Consequently, knowledge editing has emerged as a promising approach to precisely update knowledge within LLMs, while preserving performance on unrelated knowledge (Cao et al., 2024; Yao et al., 2023; Zhang et al., 2024b). In practical applications, if LLM can only parrot the edited knowledge without true understanding, it will struggle to respond correctly when the query is phrased differently. Therefore, it is crucial for LLMs to apply edited knowledge flexibly and robustly across diverse contexts. Existing knowledge editing methods can be broadly divided into parameter-preserving and parameter-modifying approaches.

(1) Parameter-preserving methods (Wang et al., 2024; Mitchell et al., 2022b; Zheng et al., 2023; Yu et al., 2024) rely on external components to steer the model's output toward edited knowledge. This approach is efficient and largely preserves the model's original parameters. However, the new knowledge does not integrate into the model's internal parameters. In practical use, users naturally pose questions in varied ways to inquire about the same fact, yet the model lacks the corresponding internal parameters to perform the semantic association and reasoning needed to handle such variations. Therefore, achieving robust generalization requires methods that fuse knowledge directly into the model's parameters. (2) Parameter-modifying methods (Mitchell et al., 2022a; Meng et al., 2022; 2023), update the model's weights directly, typically by optimizing specific layers. This allows the edited knowledge to integrate within the updated layers. Nevertheless, these methods face two key challenges. First, an edit typically establishes a direct association between a subject and its target object within the updated layers. This mapping may not cover the varied contexts encountered during generalization (Yao et al., 2023). Second, because only the targeted layers are updated,

[1] https://anonymous.4open.science/r/Edit-GRPO-119D

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.

a semantic mismatch can arise between the edited layers and the unchanged layers, which may lead to model collapse.

In response to these limitations, Reinforcement Learning (RL) offers a promising alternative for editing LLMs (Li et al., 2024b). RL frames knowledge updates as a problem of global policy optimization (Shao et al., 2024; Yu et al., 2025; Schulman et al., 2017), and provides a pathway to mitigate the key limitation of parameter-modifying methods by enabling context-based generation with network-wide knowledge integration (Li et al., 2025). However, applying RL like GRPO to generalize a single edit to diverse semantic variants still faces two challenges. First, it treats each query independently and requires multiple rollouts per query, which is inefficient and fails to capture the common edited knowledge across different semantic variants, limiting generalization to unseen samples. Second, updates on generalization queries may conflict with the original edit (Fig. 2), causing the model to drift from the edited fact.

To address these two challenges, we propose EAGPO, a framework that extends post-edit RL enabling the model to generalize a single edit to diverse semantic variants within predefined transformation categories instead of overfitting to the target expression. Specifically, it optimizes over structured query clusters with Anchor-Generalization Training (AGT) and stabilizes updates via geometric gradient alignment with the Policy Gradient Alignment Advantage (PGAA) module. First, new knowledge is precisely edited into the model by a base knowledge editing method, ensuring the model serves as a reliable anchor. Second, we initialize RL optimization from the edited model to improve semantic-variant generalization, so our core innovations lie in the RL phase. To address inefficient sampling and limited generalization caused by independent per-query rollouts, AGT constructs a semantic cluster for each edit, grouping the anchor query with its semantic variants of the same edited fact. Optimizing this cluster under the anchor constraint enables AGT to improve sampling efficiency compared with GRPO and to learn generalized semantic representations of the edited knowledge. To address unstable updates caused by baseline bias and gradient conflict, we design PGAA, which incorporates a Peer-Mean Reward Baseline and Geometric Gradient Alignment into the advantage function. Specifically, to prevent the anchor query's disproportionately high reward from inflating the baseline and distorting the policy gradient, we estimate the reward baseline solely from the generalization group, yielding an unbiased reference for optimization. For advantage estimation, Geometric Gradient Alignment guides the gradient updates for generalization queries to stay aligned with the anchor query's gradient direction. PGAA thus provides precise control over the trade-off between knowledge memorization and generalization capability. Experimental results across diverse settings consistently validate the effectiveness of our approach.

The main contributions are summarized as follows.

- On the KnowEdit benchmark, EAGPO improves generalization performance by 25%–35% over the original knowledge editing across three generalization evaluation.
- Theoretical analysis demonstrates that Policy Geometric Gradient Alignment module effectively reduces the variance of policy gradient estimation by tuning the gradient alignment coefficient.
- For efficiency, EAGPO matches the training speed of SFT while outperforming standard GRPO by 25%.
- EAGPO demonstrates robust scalability for real-world sequential editing, enabling continual knowledge updates across multiple sessions without catastrophic forgetting.

2 Related Works

2.1 Knowledge Editing in LLMs

Existing knowledge editing techniques are broadly categorized by their intervention on model parameters. Parameter-preserving methods avoid direct weight changes by employing external memory (Mitchell et al., 2022b), adapter modules (Wang et al., 2024; Yu et al., 2024), or instructional prompts (Zheng et al., 2023). Frameworks such as SERAC (Mitchell et al., 2022b) and GRACE store edits externally, while methods like IKE (Zheng et al., 2023) utilize constructed instructions to guide model behavior. This paradigm safeguards original model integrity but inherently limits generalization to the capacity of the external component. Conversely, Parameter-modifying methods

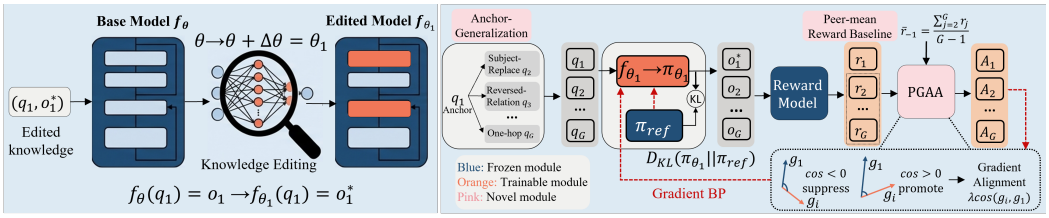

Figure 1: EAGPO framework. Left: Editing aligns anchor query q_1 with target o_1^*. Right: The edited policy undergoes RL, applying PGAA across anchor and generalization samples under a peer-mean reward baseline to enhance generalization.

directly update the model's internal weights (Mitchell et al., 2022a; De Cao et al., 2021; Meng et al., 2022; 2023; Zhang et al., 2024a; Li et al., 2024a). Early work includes MEND which learns a meta-network for updates, while KE (Mitchell et al., 2022a) and subsequent localization-based methods like ROME (Meng et al., 2022), MEMIT (Meng et al., 2023) and PMET (Li et al., 2024a) employ associative key-value updates to modify specific model facts. However, the generalization capability of key-value edits is often restricted to a narrow distribution around the edited fact, struggling with multi-hop reasoning (Fang et al., 2025b;a). Our method addresses limitations by leveraging RL to optimize the model's policy for generalized knowledge application, enabling robust compositional reasoning without model collapse.

2.2 Generalization in RL for LLMs

Achieving robust generalization in RL for LLMs remains a fundamental challenge (Bai et al., 2025; Huan et al., 2025; Yang et al., 2024). Classic policy gradient algorithms like PPO (Schulman et al., 2017) and their variants, widely used for LLM alignment, often employ regularization techniques such as entropy bonuses (Cheng et al., 2025; Wang et al., 2025) or L2 weight penalties (Aniţa & Turinici, 2024) to mitigate overfitting and improve stability. For complex reasoning, recent algorithms like GRPO (Shao et al., 2024) and its extension DAPO (Yu et al., 2025) utilize group-wise relative comparisons to efficiently optimize for reasoning paths without a value model. Despite these advances, RL optimization is inherently susceptible to policy fragility, where minor reward shifts can cause drastic policy changes (Xu, 2025), leading to high-variance outcomes and compromising generalization. This instability is particularly detrimental in knowledge editing, where the optimization must remain anchored to a precise factual update (Li et al., 2025; Nguyen et al., 2026). Notably, RL has been applied to lifelong editing scalability, with methods RLEdit (Li et al., 2024b) focuses on optimizing a hypernetwork for sequential editing via RL, rather than for generalization. EtCon (Li et al., 2025) employs GRPO solely for consolidation to align parametric knowledge with generation behavior. Our EAGPO framework introduces a gradient-alignment mechanism that acts as a geometric regularizer, effectively smoothing the policy optimization trajectory around the edit anchor to ensure stable and controlled generalization.

3 Methods

3.1 Preliminary

Objective of Knowledge Editing. For editing operation on a LLM f_θ with parameter θ, given the new knowledge (query q_1, object o_1^*), knowledge editing aims to integrate new editing knowledge while preserving previous edited knowledge and maintaining general abilities. The editing operation is formulated as $\theta_1 = \mathcal{E}(\theta, q_1, o_1^*)$, which updates the model parameters from θ to θ_1 and correspondingly evolves the LLM from f_θ to f_{θ_1}. We have $f_{\theta_1}(q_1) = o_1^*$. For data points $\bar{x}$ not associated with the editing data, the LLM should retain the original predictions, ensuring that $f_{\theta_1}(\bar{x}) \approx f_\theta(\bar{x})$. This defines the objectives of reliability and locality for a single edit. However, a successful edit should enable the model to generalize the editing knowledge to related queries and reasoning tasks, which is the focus of our work.

Group Relative Policy Optimization (GRPO). GRPO eliminates the external critic by normalizing rewards within a sampled group. For each query q, it samples G outputs $\{o_i\}_{i=1}^{G}$ from π_θ. The

advantage is $A_i = \frac{r_i - \bar{r}}{\sigma}$, with $\bar{r} = \frac{1}{G}\sum_{i=1}^{G} r_i$ and $\sigma = \frac{1}{G}\sum_{j=1}^{G}(r_j - \bar{r})^2$. This group-relative evaluation provides objective-specific signals suited for knowledge editing. However, the global mean may be biased by high-reward samples, which motivates our adjustment in the next section.

3.2 EAGPO Framework

EAGPO framework is illustrated in Fig. 1. First, the model performs original knowledge editing. Then the edited model with updated knowledge serves as the starting point for the RL phase. The RL phase comprises two main components: (i) Anchor-Generalization Training (AGT), which trains the model on a cluster of edited knowledge and semantic variants under the anchor query's central constraint. (ii) The Policy Gradient Alignment Advantage (PGAA) module, which stabilizes optimization by imposing a gradient consistency constraint from generalization data on the advantage estimation.

3.3 Knowledge Editing

To provide the policy model with edited knowledge and anchor it to the target fact, we first apply a knowledge editing method to inject the fact directly into the model parameters. This foundational step enables the subsequent RL phase to concentrate solely on generalizing across diverse semantic variants, effectively preventing optimization drift away from the correct knowledge and thereby facilitating robust generalization to related edits.

Formally, given a LLM f_θ and editing knowledge (q_1, o_1^*), as described in Preliminary 3.1, following the editing operation we obtain the post-edit model f_{θ_1} with parameters θ_1.

$$\theta_1 = \mathcal{E}(\theta, q_1, o_1^*), \quad \text{s.t. } f_{\theta_1}(q_1) = o_1^*, \tag{1}$$

where $\mathcal{E}(\cdot)$ denotes the original knowledge editing algorithm. This operation updates the model from θ to θ_1, ensuring that the post-edit model reliably incorporates the new fact. Crucially, f_{θ_1} serves as a starting policy model π_{θ_1} subsequent RL in EAGPO. :

$$\pi_{\theta_1}(q) \triangleq f_{\theta_1}(q), \quad \forall q \in \mathcal{Q}, \tag{2}$$

where $\mathcal{Q}$ denotes the space of all possible queries. By editing the initial knowledge, knowledge editing provides the necessary semantic foundation for the RL to effectively generalize the knowledge.

3.4 Anchor-Generalization Training

To address the inefficiency and limited generalization caused by independent per-query optimization with multiple rollouts in RL, we treat the anchor query and its semantic variants as a shared cluster, so that the model is encouraged to capture the common edited knowledge across different phrasings. As illustrated in Fig. 1, the cluster comprises the anchor query q_1 and generalization queries $\mathcal{Q}_{\text{gen}} = \{q_2, \cdots, q_G\}$ sharing the same edited knowledge. The anchor query then serves as a constraining center for policy optimization, forming our Anchor-Generalization Training (AGT) framework.

Specifically, we construct a query cluster for the edited knowledge containing $|G|$ samples: $\mathcal{B} = \{q_1\} \cup \mathcal{Q}_{\text{gen}} = \{q_1, \cdots, q_G\}$. This cluster consists of two parts: the Anchor Query q_1, which corresponds to the precise fact edited, and the generalization query group $\mathcal{Q}_{\text{gen}}$, which contains $G-1$ varied expressions of the same knowledge from the training set. This entire query cluster $\mathcal{B}$ is processed through the post-edit policy model π_{θ_1} after knowledge editing, yielding a corresponding set of responses $\{o_1^*, \cdots, o_G\}$. The objective function aims to maximize the expected advantage while applying a KL divergence to prevent the policy from drifting too far from the reference model π_{ref}:

$$\mathcal{J}_{\text{GRPO}}(\theta_1) = \frac{1}{G}\sum_{i=1}^{G} \min\left(\frac{\pi_{\theta_1}(o_i|q_i)}{\pi_{\theta_{\text{old}}}(o_i|q_i)} A_i, \right.$$
$$\left. \text{clip}\left(\frac{\pi_{\theta_1}(o_i|q_i)}{\pi_{\theta_{\text{old}}}(o_i|q_i)}, 1-\epsilon, 1+\epsilon \right) A_i \right) - \beta \mathbb{D}_{\text{KL}}\big(\pi_{\theta_1} \parallel \pi_{\text{ref}}\big), \tag{3}$$

where β controls the strength of the KL penalty, A_i is the relative advantage of RL, $\pi_{\theta_{old}}$ denote the policy parameters before the update and ϵ is the clipping range hyperparameter.

Unlike GRPO's multiple rollouts per sample, our method uses a single rollout per sample but groups the anchor with its variants in a batch. This joint optimization propagates gradients to a shared cluster representation, enabling inherent generalization to all in-cluster queries without redundant sampling.

3.5 Policy Gradient Alignment Advantage

To keep policy updates on generalization queries consistent with the edited knowledge, we keeps policy updates on generalization queries aligned with the original edit, preserving knowledge consistency within the local parameter geometry. After obtaining responses $\{o_1, \cdots, o_G\}$, we compute rewards and derive advantages based on two core designs: Peer-Mean Reward Baseline and Geometric Gradient Alignment.

Peer-Mean Reward Baseline. To prevent the anchor query's reward from making the mean reward too high, we derive the baseline solely from the generalization group. Since the policy π_{θ_1} is already edited on q_1, it typically receives r_1 that far exceeds generalization rewards. Using GRPO normalization with the entire group's mean including r_1 would underestimate the advantage for generalization samples, and this bias undermines stable policy updates. To eliminate this, we propose a Peer-Mean Reward Baseline that subtracts the mean of its peer queries in $\mathcal{Q}_{\text{gen}}$. Formally, we compute this mean and variance for advantage computation, and following the GRPO principle of using normalized reward as the advantage value, we first compute a preliminary advantage for each sample as:

$$\bar{r}_{-1} = \frac{1}{G-1}\sum_{j=2}^{G} r_j, \quad \sigma_{-1}^2 = \frac{1}{G-1}\sum_{j=2}^{G}(r_j - \bar{r}_{-1})^2, \quad A_i' = \frac{r_i - \bar{r}_{-1}}{\sigma_{-1}} \tag{4}$$

The calculation of rewards is described in Implementation Details (Appendix A). This design ensures comparable reward scales within the query cluster. Crucially, this normalization prevents scale differences from distorting gradient alignment by expressing advantages in comparable units relative to the cluster mean and standard deviation.

Geometric Gradient Alignment. Building on the Peer-Mean Reward Baseline, we introduce Geometric Gradient Alignment to enforce consistent update directions across generalization queries. Naive training on generalization samples can produce gradients that conflict with the anchor gradient (Fig. 2), causing the policy to drift from the edited knowledge. To mitigate this, we align the policy gradient of $\mathcal{Q}_{\text{gen}}$ with the anchor query q_1, ensuring knowledge consistency within the local parameter geometry.

For each query q_i, the policy gradient is $g_i = \nabla_\theta \log \pi_\theta(o_i|q_i)$. For any generalization query q_j $(j \geq 2)$, the anchor gradient serves as the reference direction. The advantage is then geometrically modulated by directional agreement, favoring updates that stay within the local edited region while damping divergent ones. The final advantage is then defined as:

$$A_j = A_j' + \lambda \cdot \cos(\boldsymbol{g_j}, \boldsymbol{g_1}), \quad \cos(\boldsymbol{g_j}, \boldsymbol{g_1}) = \frac{< \boldsymbol{g_j}, \boldsymbol{g_1} >}{\|\boldsymbol{g_j}\|\|\boldsymbol{g_1}\|}.$$

Here, we detach $\cos(\boldsymbol{g}_j, \boldsymbol{g}_1)$ from the computation graph, treating A_j as a constant coefficient during policy gradient updates. Applied to the specific layer following Sec.5.4, the coefficient $\lambda > 0$ modulates advantages geometrically. For the anchor q_1, we retain $A_1 = A_1'$ to protect the edited knowledge. For generalization samples, alignment with the anchor ($\cos > 0$) amplifies A_j above A_j', while conflict ($\cos < 0$) suppresses it below A_j'. This geometric modulation ensures stable generalization. Algorithm pseudocode is in Appendix.

4 Theoretical Analysis

Theoretically, we prove our method better protects the edited knowledge, improves generalization, and achieves a lower convergence error upper bound. Proofs of all theorems are provided in Appendix B.

Theorem 4.1 (Reliability of EAGPO). *For EAGPO with $\lambda > 0$ and step size η, let $g_1 = \nabla_\theta \log \pi_\theta(o_1^*|q_1)$ and $g_j = \nabla_\theta \log \pi_\theta(o_j|q_j)$ $(j = 2, \ldots, G)$. Define $\alpha = \sum_{j=2}^{G} \frac{\langle g_j, g_1 \rangle^2}{\|g_j\|\|g_1\|} > 0$*

and $\kappa = \sum_{j=1}^{G}(r_j - \bar{r}_{-1})\langle g_1, g_j\rangle$*, where* $\bar{r}_{-1}$ *is the mean reward of the generalization group. The anchor log-likelihood change after one update satisfies* $\Delta\mathcal{L}_1 = \eta(\lambda\alpha + \kappa) + O(\eta^2)$*. Thus,* $\Delta\mathcal{L}_1 > 0$ *is guaranteed for any* $\lambda > \lambda_0 := \max(0, -\kappa/\alpha)$*.*

Thm. 4.1 ensures reliability by increasing anchor log-likelihood via a gradient update combining reward and alignment terms with a proper λ.

Theorem 4.2 (Generalization of EAGPO)**.** *Assume loss* ℓ *is* L*-Lipschitz and* μ*-smooth, rewards are bounded by* C*, and two datasets differ only in a fixed anchor sample. Define the conflict set* $\Omega(\theta) = \{i \in \{2, \ldots, G\} \mid (r_i - \bar{r}_{-1}) \cdot \cos\left(\nabla_\theta \ell(\theta; z_i), g_1(\theta)\right) < 0\}$ *and the average conflict strength* $\rho(\theta) = \frac{1}{|\Omega|}\sum_{i\in\Omega} |\cos\left(\nabla_\theta \ell(\theta; z_i), g_1(\theta)\right)|$*. With step size* $\eta \leq 1/\mu$*, EAGPO's uniform stability constant is:*

$$\beta_{edit} \leq \left(1 - \frac{\lambda\, \rho\, |\Omega|}{2CG}\right) \beta_{std} + \mathcal{O}(\eta^2).$$

This implies ϵ_{gen}*(EAGPO)* $< \epsilon_{gen}$*(std-GRPO) under conflicting samples.*

Thm. 4.2 proves mitigating conflicting gradients yields a smaller stability constant and lower generalization error for EAGPO than GRPO.

Assumption 4.3 (Reward Distribution After Editing)**.** For a batch $\mathcal{S}$ with successful anchor editing, we assume: (i) $\mathbb{E}[r_1] = \mu_1$ and $\mathbb{E}[r_i] = \mu_g$ for $i = 2, \ldots, G$, with $\mu_1 > \mu_g$; (ii) all rewards are bounded, $|r_i| \leq C$; and (iii) rewards are conditionally independent given the queries, with $\mathrm{Var}(r_i) = \sigma^2$ for $i > 1$.

Assumption 4.4 (Gradient Properties)**.** The score functions $\boldsymbol{g}_i = \nabla_\theta \log \pi_\theta(o_i|q_i)$ satisfy: (i) $\|\boldsymbol{g}_i\| \leq L$ for all i; (ii) the anchor gradient $\boldsymbol{g}_1$ is fixed relative to the batch randomness $\{r_i\}_{i=2}^G$ and $\{\boldsymbol{g}_i\}_{i=2}^G$; (iii) the conflict set $\Omega = \{i \in \{2, \ldots, G\} : \cos(\boldsymbol{g}_i, \boldsymbol{g}_1) < 0\}$ satisfies $|\Omega| > 0$, with average squared cosine similarity $\rho^2 = \frac{1}{|\Omega|}\sum_{i\in\Omega} \cos^2(\boldsymbol{g}_i, \boldsymbol{g}_1) > 0$; and (iv) $\|\boldsymbol{g}_i\| \geq L_{\min} > 0$ for all $i \in \Omega$.

Theorem 4.5 (Variance of EAGPO)**.** *Under Assumptions 4.3 and 4.4, for a successful edit (*$\mu_1 > \mu_g$*) with conflict set* $|\Omega| = \gamma G > 0$*, define*

$$\Gamma = \sqrt{\frac{\gamma L_{\min}^2 (\mu_1 - \mu_g)^2}{\gamma L_{\min}^2 \rho^2 + L^2(1-\gamma)}}.$$

If $0 < \lambda < \frac{\Gamma}{G}$*, then variance of EAGPO's estimator satisfies* $Var(\hat{\boldsymbol{g}}_{edit}) < Var(\hat{\boldsymbol{g}}_{std})$*.*

Thm. 4.5 proves a suitable λ reduces policy gradient variance below GRPO under conflicting samples, ensuring stable optimization.

The theoretical analysis informs an adaptive λ schedule: starting low to promote stable generalization and increasing midway to consolidate the edit. This strategy is empirically validated by the sensitivity analysis in Tab. 4 and variance reduction in Fig. 2.

5 Experiments

We provide full details regarding the experimental setup, including datasets, evaluation metrics, implementation details and baselines, in the Appendix A.

5.1 Main Results and Analysis

We evaluate EAGPO on Qwen3-8B and Llama3-8B using the KnowEdit benchmark on unseen generalization samples. Tab. 1 reports Qwen3-8B on reliability, three generalization scenarios, and locality, with Llama3-8B results in Appendix A.4.

Failure of Direct Injection and Model Collapse. *Initial Editing* with only q_1 ensures high reliability but exhibits brittle generalization. The *All Editing* strategy injecting $q_1 \sim q_G$ triggers catastrophic weight shifts and model collapse for parameter modifying methods. Parameter preserving methods like WISE avoid total collapse due to side memory but fail to achieve optimal generalization compared to EAGPO.

Table 1: Knowledge editing performance of Qwen3-8B on the KnowEdit benchmark. We report accuracy (%) across four dimensions: **Reliability** (↑), **Subject-Replace** (↑), **Reverse-Relation** (↑), and **One-hop** (↑) for generalization, and **Locality** (↑) for knowledge retention. Bold indicates the best performance. Underline indicates the second-best performance.

Editor	Gen. phase	Rel.	S-R.	R-R.	O-H.	Loc.
ROME	Initial Editing	**98.85±0.12**	30.52±0.28	50.31±0.35	53.17±0.42	**68.24±0.28**
	All Editing ($q_{1\sim G}$)	22.45±0.32	18.61±0.14	20.18±0.41	20.23±0.46	18.35±0.32
	SFT Baseline	92.15±0.41	86.27±0.39	83.82±0.47	65.96±0.51	51.42±0.53
	GRPO (EtCon)	94.35±0.22	83.18±0.12	82.42±0.52	70.47±0.56	66.28±0.28
	EAGPO (Ours)	97.85±0.15	**88.35±0.18**	**85.64±0.43**	**72.78±0.15**	67.73±0.24
MEMIT	Initial Editing	**99.10±0.08**	43.23±0.31	49.03±0.41	55.82±0.38	70.87±0.32
	All Editing ($q_{1\sim G}$)	24.82±0.41	20.57±0.26	25.76±0.39	20.94±0.44	20.72±0.41
	SFT Baseline	94.52±0.33	86.83±0.21	83.32±0.44	63.34±0.49	54.19±0.51
	GRPOO (EtCon)	95.88±0.45	86.92±0.36	85.16±0.28	67.83±0.54	68.64±0.35
	EAGPO (Ours)	98.24±0.18	**89.84±0.21**	**87.28±0.29**	**71.42±0.36**	**71.15±0.32**
MEND	Initial Editing	**94.52±0.35**	40.08±0.29	1.47±0.12	13.49±0.36	**62.05±0.35**
	All Editing ($q_{1\sim G}$)	18.35±0.28	15.14±0.31	1.89±0.21	1.82±0.42	4.18±0.05
	SFT Baseline	89.45±0.52	84.61±0.38	65.32±0.26	44.85±0.48	45.76±0.26
	GRPO (EtCon)	90.12±0.55	83.27±0.44	66.18±0.34	62.73±0.52	58.17±0.31
	EAGPO (Ours)	93.56±0.32	**85.85±0.29**	**69.87±0.44**	**67.96±0.47**	61.58±0.27
WISE	Initial Editing	98.45±0.22	50.84±0.32	48.19±0.27	45.07±0.39	**74.12±0.30**
	All Editing ($q_{1\sim G}$)	96.12±0.45	82.28±0.38	84.45±0.35	57.29±0.45	70.23±0.38
	SFT Baseline	95.82±0.25	90.45±0.15	82.17±0.38	55.86±0.41	59.32±0.18
	GRPO (EtCon)	96.25±0.31	89.82±0.20	88.35±0.16	65.64±0.68	73.05±0.25
	EAGPO (Ours)	**98.85±0.12**	**91.68±0.11**	**90.42±0.38**	**69.33±0.22**	73.86±0.19
AlphaEdit	Initial Editing	98.21±0.15	43.06±0.21	51.28±0.33	56.18±0.25	**69.43±0.25**
	All Editing ($q_{1\sim G}$)	25.67±0.33	21.13±0.10	25.42±0.40	21.56±0.36	23.15±0.32
	SFT Baseline	96.12±0.35	90.59±0.23	83.24±0.42	66.73±1.18	54.43±0.64
	GRPO (EtCon)	96.54±0.42	89.46±0.37	86.19±0.49	70.27±0.75	68.28±0.22
	EAGPO (Ours)	**98.92±0.14**	**92.72±0.20**	**88.18±0.35**	**71.14±0.19**	67.21±0.16
InstructEdit	Initial Editing	97.68±0.18	60.83±0.12	57.85±0.30	56.56±0.26	**70.15±0.22**
	All Editing ($q_{1\sim G}$)	24.91±0.25	20.57±0.20	2.42±0.10	1.56±0.06	20.15±0.12
	SFT Baseline	97.25±0.21	91.82±0.18	77.45±0.28	68.12±0.34	55.34±0.25
	GRPO (EtCon)	97.15±0.28	91.24±0.25	79.87±0.32	72.15±0.41	67.82±0.18
	EAGPO (Ours)	**99.52±0.10**	**94.12±0.14**	**82.35±0.22**	**77.56±0.28**	68.92±0.12

SFT and GRPO Baselines. *SFT* boosts Reliability via rote memorization, achieving high Subject Replace scores, but severely impairs Locality. *GRPO* (EtCon) promotes better logical exploration and higher One hop scores, yet without anchoring constraints it suffers from high variance in Reliability due to occasional drift from the factual anchor.

Effectiveness of EAGPO. *EAGPO* resolves the stability-plasticity dilemma via PGAA, achieving 77.56% one-hop generalization with InstructEdit and 99.52% reliability. Unlike SFT, it preserves locality through a geometric regularizer that confines updates to the edit's neighborhood, protecting unrelated knowledge.

5.2 Ablation Study

The ablation study presented in Tab. 2 validates the effectiveness of the core components in the EAGPO framework particularly in the more challenging *One-hop* and *Locality* generalization metrics. Removing PGAA ($\lambda = 0$) drops One-hop by 3.59%, while omitting $\bar{r}_{-1}$ or the edit anchor also degrades performance, with further reward model ablations provided in Appendix A.8. Data ablations show excluding q_1 reduces Locality by 4.30%, and One-by-one training underperforms cluster-based AGT. Overall, PGAA is the most critical component, corroborating our theory that geometric alignment balances knowledge memorization and generalization, yielding lower gradient variance and a tighter bound.

Table 2: Ablation study of EAGPO with **MEMIT** editor on the KnowEdit dataset.

Variant	Rel.	S-R.	R-R.	O-H.	Loc.
EAGPO	**98.24**	**89.84**	**87.28**	**71.42**	**71.15**
Module Ablation					
w/o PGAA ($\lambda = 0$)	96.15	86.92	85.16	67.83	70.64
w/o $\bar{r}_{-1}$ Baseline	97.62	88.51	86.42	69.15	70.28
w/o knowledge editing	82.62	80.51	77.42	65.15	71.10
Data Input Ablation					
Gen Data Only (w/o q_1)	93.35	85.14	84.95	67.38	66.85
One-by-one	96.88	86.45	85.32	69.04	70.32

5.3 Real-World Sequential Editing with EAGPO

We evaluate EAGPO's long-term stability on the real-world WikiBigEdit benchmark via five sequential editing sessions, cumulatively updating 10,000 facts with AlphaEdit as the editor. As shown in Tab. 3, the framework remains remarkably stable, with only a 2.78% decline in Reliability and a 4.93% drop in One-hop after five sessions. This contrasts sharply with the catastrophic collapse observed in serial MEMIT/ROME under All Editing. We find that for these methods, sequential editing causes a sharp increase in the norm of the edited layers, which leads to model collapse and makes further editing infeasible. In contrast, EAGPO's robustness stems from AlphaEdit's minimally invasive projections and PGAA's geometric regularization, which aligns new generalization gradients with anchors to prevent overwriting prior knowledge, confirming EAGPO as a scalable and sustainable framework for continual model updates without full retraining.

Table 3: Sequential editing performance across five sessions using EAGPO.

Edit Session	Rel.	S-R.	R-R.	O-H.	Loc.
E_1	98.02	92.54	90.58	73.65	68.21
E_2	97.65	91.42	89.17	72.10	67.97
E_3	96.98	89.65	87.54	71.24	65.12
E_4	96.15	87.26	86.13	69.13	64.07
E_5	95.24	87.18	85.82	68.72	63.60

5.4 Gradient Alignment Analysis

Gradient Alignment and Training Stability. Fig. 2 compares EAGPO and GRPO. Left: EAGPO reaches 0.7 cosine similarity, confirming PGAA minimizes $\Omega(\theta)$ and internalizes edits. Right: variance drops to a low stable level within 150 steps, whereas GRPO remains at a higher level. This validates that gradient alignment regularizes optimization and ensures stable updates. Overall, the results confirm that EAGPO achieves a robust balance between knowledge retention and stable policy optimization.

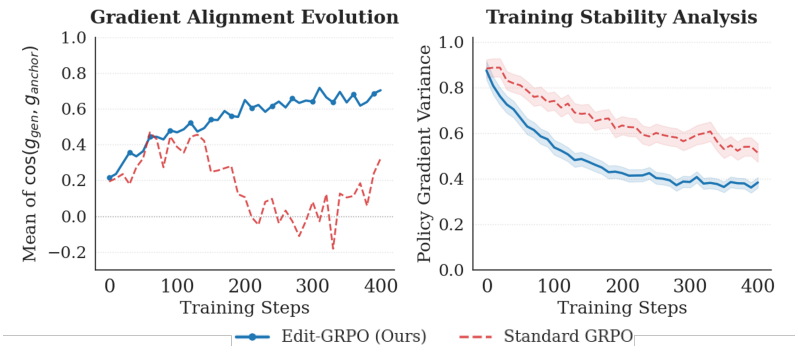

Figure 2: Gradient Alignment and Variance Evolution

Table 4: Sensitivity of the alignment coefficient λ for EAGPO with editor MEMIT, dynamically annealed from 0.01 to the tabulated values at the training midpoint.

Coefficient λ	Rel.	S-R.	R-R.	O-H.	Loc.
0.05	96.85	87.02	85.34	68.25	70.85
0.10	97.42	88.95	86.41	70.12	70.42
0.20 (Ours)	98.24	89.84	**87.28**	**71.42**	**71.15**
0.30	98.65	89.12	86.56	69.74	69.04
0.50	**99.02**	**90.45**	86.18	65.20	67.58

Impact of Alignment Depth. Fig. 3 analyzes alignment layer sensitivity. Locate-then-Edit methods peak at initial editing layers, confirming local reinforcement, whereas InstructEdit and WISE exhibit robustness with a slight preference for the final layer. This validates our adaptive strategy of selecting the edit layer for localized editors and the final layer for global editors, ensuring optimal cross-query consistency.

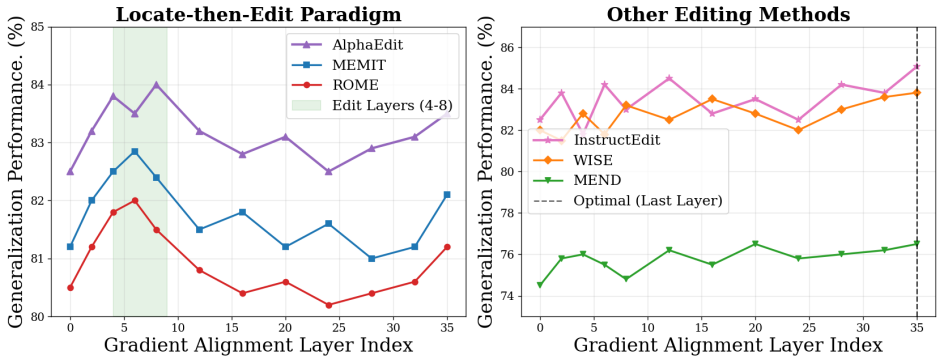

Figure 3: Impact of alignment layer selection on generalization performance.

Impact of Alignment Coefficient λ. Tab. 4 reveals a generalization-reliability trade-off for λ. Lower values improve One-hop reasoning, while $\lambda = 0.50$ maximizes Reliability. This aligns with Thm. 4.5 for stability and Thm. 4.1 for core edit protection. Our chosen $\lambda = 0.20$ effectively balances strong Reliability with competitive generalization across all reasoning forms.

5.5 Other Experiments

We further analyze EAGPO through Appendix. It includes Llama3-8B performance (A.4), computation efficiency (A.5), robustness validation on downstream GLUE evaluation (A.6), multi-hop reasoning evaluation on MQuAKE, generalization transfer across different relation and templates (A.7), reward model ablation (A.8) and p-values significance tests (A.9). We also present a case study (C) illustrating different type gradient-reward conflicts, demonstrating how our framework guides optimization toward self-consistent generalization.

6 Conclusion

We introduced EAGPO framework to address the poor generalization in knowledge editing via RL. Our method combines an initial knowledge editing with RL-based generalization. RL is achieved through two novel components: AGT which structures learning around query clusters, and PGAA which stabilizes updates via geometric gradient alignment. Theoretically, we prove that EAGPO provides a lower generalization error bound and reduces policy gradient variance. Empirically, it achieves significant gains over strong baselines on the KnowEdit benchmark, all while maintaining training efficiency comparable to SFT. This work provides a principled and effective pathway towards editable language models that are both precise and broadly applicable.

REFERENCES

Ştefana-Lucia Aniţa and Gabriel Turinici. Convergence of a l2 regularized policy gradient algorithm for the multi armed bandit. In *International Conference on Pattern Recognition*, pp. 407–422. Springer, 2024.

Haoyue Bai, Yiyou Sun, Wenjie Hu, Shi Qiu, Maggie Ziyu Huan, Peiyang Song, Robert Nowak, and Dawn Song. How and why llms generalize: A fine-grained analysis of llm reasoning from cognitive behaviors to low-level patterns. *arXiv preprint arXiv:2512.24063*, 2025.

Boxi Cao, Hongyu Lin, Xianpei Han, and Le Sun. The life cycle of knowledge in big language models: A survey. *Machine Intelligence Research*, 21(2):217–238, 2024.

Daixuan Cheng, Shaohan Huang, Xuekai Zhu, Bo Dai, Wayne Xin Zhao, Zhenliang Zhang, and Furu Wei. Reasoning with exploration: An entropy perspective. *arXiv preprint arXiv:2506.14758*, 2025.

Nicola De Cao, Wilker Aziz, and Ivan Titov. Editing factual knowledge in language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 6491–6506, 2021.

Junfeng Fang, Houcheng Jiang, Kun Wang, Yunshan Ma, Xiang Wang, Xiangnan He, and Tat-seng Chua. Alphaedit: Null-space constrained knowledge editing for language models. *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025a.

Quntian Fang, Zhen Huang, Zhiliang Tian, Minghao Hu, Dongsheng Li, Yiping Yao, Xinyue Fang, Menglong Lu, and Guotong Geng. Hippocampal-like sequential editing for continual knowledge updates in large language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025b.

Maggie Huan, Yuetai Li, Tuney Zheng, Xiaoyu Xu, Seungone Kim, Minxin Du, Radha Poovendran, Graham Neubig, and Xiang Yue. Does math reasoning improve general llm capabilities? understanding transferability of llm reasoning. *arXiv preprint arXiv:2507.00432*, 2025.

Ruilin Li, Yibin Wang, Wenhong Zhu, Chenglin Li, Jinghao Zhang, Chenliang Li, Junchi Yan, and Jiaqi Wang. Etcon: Edit-then-consolidate for reliable knowledge editing. *arXiv preprint arXiv:2512.04753*, 2025.

Xiaopeng Li, Shasha Li, Shezheng Song, Jing Yang, Jun Ma, and Jie Yu. Pmet: Precise model editing in a transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 18564–18572, 2024a.

Zherui Li, Houcheng Jiang, Hao Chen, Baolong Bi, Zhenhong Zhou, Fei Sun, Junfeng Fang, and Xiang Wang. Reinforced lifelong editing for language models. In *Forty-second International Conference on Machine Learning*, 2024b.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372, 2022.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. Mass editing memory in a transformer. *The Eleventh International Conference on Learning Representations (ICLR)*, 2023.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. Fast model editing at scale. In *The Tenth International Conference on Learning Representations (ICLR)*, 2022a.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D Manning, and Chelsea Finn. Memory-based model editing at scale. In *International Conference on Machine Learning*, pp. 15817–15831. PMLR, 2022b.

Duy Nguyen, Hanqi Xiao, Archiki Prasad, Elias Stengel-Eskin, Hyunji Lee, and Mohit Bansal. Conflict-resolving and sharpness-aware minimization for generalized knowledge editing with multiple updates. *arXiv preprint arXiv:2602.03696*, 2026.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Lukas Thede, Karsten Roth, Matthias Bethge, Zeynep Akata, and Thomas Hartvigsen. Wikibigedit: Understanding the limits of lifelong knowledge editing in llms. In *International Conference on Machine Learning*, pp. 59326–59354. PMLR, 2025.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 353–355, 2018.

Peng Wang, Zexi Li, Ningyu Zhang, Ziwen Xu, Yunzhi Yao, Yong Jiang, Pengjun Xie, Fei Huang, and Huajun Chen. Wise: Rethinking the knowledge memory for lifelong model editing of large language models. *Advances in Neural Information Processing Systems*, 37:53764–53797, 2024.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, et al. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *arXiv preprint arXiv:2506.01939*, 2025.

Xingcheng Xu. The policy cliff: A theoretical analysis of reward-policy maps in large language models. *arXiv preprint arXiv:2507.20150*, 2025.

Rui Yang, Ruomeng Ding, Yong Lin, Huan Zhang, and Tong Zhang. Regularizing hidden states enables learning generalizable reward model for llms. *Advances in Neural Information Processing Systems*, 37:62279–62309, 2024.

Yunzhi Yao, Peng Wang, Bozhong Tian, Siyuan Cheng, Zhoubo Li, Shumin Deng, Huajun Chen, and Ningyu Zhang. Editing large language models: Problems, methods, and opportunities. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 10222–10240, 2023.

Lang Yu, Qin Chen, Jie Zhou, and Liang He. Melo: Enhancing model editing with neuron-indexed dynamic lora. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19449–19457, 2024.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Ningyu Zhang, Bozhong Tian, Siyuan Cheng, Xiaozhuan Liang, Yi Hu, Kouying Xue, Yanjie Gou, Xi Chen, and Huajun Chen. Instructedit: instruction-based knowledge editing for large language models. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pp. 6633–6641, 2024a.

Ningyu Zhang, Yunzhi Yao, Bozhong Tian, Peng Wang, Shumin Deng, Mengru Wang, Zekun Xi, Shengyu Mao, Jintian Zhang, Yuansheng Ni, et al. A comprehensive study of knowledge editing for large language models. *arXiv preprint arXiv:2401.01286*, 2024b.

Ce Zheng, Lei Li, Qingxiu Dong, Yuxuan Fan, Zhiyong Wu, Jingjing Xu, and Baobao Chang. Can we edit factual knowledge by in-context learning? In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 4862–4876, 2023.

Zexuan Zhong, Zhengxuan Wu, Christopher D Manning, Christopher Potts, and Danqi Chen. Mquake: Assessing knowledge editing in language models via multi-hop questions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 15686–15702, 2023.

NOTATION AND ALGORITHM

This section provides a comprehensive summary of the mathematical notations used throughout the paper. The symbols are grouped thematically to facilitate understanding of our method and theoretical analysis. Readers may pay particular attention to the symbols related to the *gradient geometry* (e.g., $\boldsymbol{g}_1$, Ω, ρ) and the *algorithm parameters* (e.g., λ, Γ), as they are central to the design and theoretical guarantees of the proposed EAGPO framework. The key notations used throughout this paper are summarized in Tab. 5.

Based on the notations summarized above, we present the complete pseudocode of EAGPO in Algorithm 1, which concretely instantiates the two-stage procedure and the key components including the anchor-generalization training, the peer-mean baseline, and the geometric gradient alignment using λ and $\cos(\boldsymbol{g}_j, \boldsymbol{g}_1)$.

Table 5: Summary of key notations.

Symbol	Description
Dataset and Batch	
$\mathcal{D}$	Underlying data distribution.
$\mathcal{S}$	A training batch of G samples, $\mathcal{S} = \{z_1, \ldots, z_G\}$.
$z_i = (q_i, o_i)$	A data sample, consisting of a query q_i and a target answer o_i.
G	Batch size.
q_1, o_1^*	The *anchor* query and its target object.
$\mathcal{Q}_{\text{gen}}$	The set of *generalization queries* within a batch, $\{q_2, \ldots, q_G\}$.
Ω	The *conflict set*: $\{i \in \{2, \ldots, G\} \mid \cos(\boldsymbol{g}_i, \boldsymbol{g}_1) < 0\}$.
γ	Conflict ratio, $\gamma = \lvert\Omega\rvert / G$.
Reward and Objective	
r_i	Reward for the i-th sample
$\bar{r}$	Average reward over the entire batch: $\frac{1}{G}\sum_{i=1}^{G} r_i$.
$\bar{r}_{-1}$	Average reward over the generalization group: $\frac{1}{G-1}\sum_{j=2}^{G} r_j$.
μ_1, μ_g	Expected reward for the anchor and a generalization sample (Assumption B.3).
C	Upper bound for the absolute reward: $\lvert r_i\rvert \le C$.
σ^2	Variance of rewards for generalization samples.
$J(\theta)$	Expected policy objective (composite reward).
$\hat{\boldsymbol{g}}_{\text{std}}, \hat{\boldsymbol{g}}_{\text{edit}}$	Gradient estimators of GRPO and EAGPO.
Gradients and Geometry	
π_θ	Policy (language model) parameterized by θ.
$\boldsymbol{g}_i$	Policy score function (gradient): $\nabla_\theta \log \pi_\theta(o_i \mid q_i)$.
$\boldsymbol{g}_1$	Anchor gradient, treated as a fixed reference direction.
L	Upper bound for the gradient norm: $\lVert\boldsymbol{g}_i\rVert \le L$ for all i.
$L_{\min}$	Lower bound for gradient norms within the conflict set: $\lVert\boldsymbol{g}_i\rVert \ge L_{\min} > 0$ for $i \in \Omega$ (Assumption B.4).
$\cos(\boldsymbol{g}_i, \boldsymbol{g}_1)$	Cosine similarity between gradients $\boldsymbol{g}_i$ and $\boldsymbol{g}_1$.
ρ	Root mean square cosine similarity in Ω: $\rho = \sqrt{\frac{1}{\lvert\Omega\rvert}\sum_{i\in\Omega}\cos^2(\boldsymbol{g}_i, \boldsymbol{g}_1)}$.
$\delta, \bar{\delta}^2$	$\delta = (r_1 - \bar{r}_{-1})/G$ and its expected square, $\bar{\delta}^2 = \mathbb{E}[\delta^2]$.
Algorithm Parameters & Theoretical Quantities	
η	Learning rate (step size).
λ	Alignment coefficient in EAGPO's advantage function.
Γ	Upper bound constant for λ: $\lambda < \Gamma/G$ (Theorem 4.2).
$\beta_{\text{std}}, \beta_{\text{edit}}$	Uniform stability constants for GRPO and EAGPO.
ϵ_{gen}	Generalization error.

Algorithm 1 EAGPO: Enhancing Generalization in Knowledge Editing

Input: LLM f_θ with initial parameters θ; editing knowledge (q_1, o_1^*); generalization query set $\mathcal{Q}_{\text{gen}} = \{q_2, \ldots, q_G\}$; knowledge editing algorithm $\mathcal{E}(\cdot)$; GRPO clipping range ϵ, KL coefficient β; RL learning rate η, alignment coefficient λ (with annealing schedule); number of training epochs N.
Output: Edited and generalized policy π_{θ^*}.
Knowledge Editing (Anchoring)
$\theta_1 \leftarrow \mathcal{E}(\theta, q_1, o_1^*)$
$\pi_{\theta_1}(q) \triangleq f_{\theta_1}(q), \quad \forall q \in \mathcal{Q}$
$\pi_{\text{ref}} \leftarrow \pi_{\theta_1}$, $\pi_{\theta_{\text{old}}} \leftarrow \pi_{\theta_1}$
$\theta \leftarrow \theta_1$
Reinforcement Learning for Generalization
for $epoch = 1$ **to** N **do**
  Construct query cluster $\mathcal{B} = \{q_1\} \cup \mathcal{Q}_{\text{gen}} = \{q_1, \ldots, q_G\}$.
  for $i = 1$ **to** G **do**
    Sample response $o_i \sim \pi_{\theta_{\text{old}}}(\cdot \mid q_i)$.
    Compute reward $r_i = \max_{a \in \mathcal{A}} \text{Jaccard}(o_i, a)$
    Compute policy gradient $g_i = \nabla_\theta \log \pi_\theta(o_i \mid q_i)$.
  end for
  Peer-Mean Reward Baseline:
  $\bar{r}_{-1} \leftarrow \frac{1}{G-1} \sum_{j=2}^{G} r_j, \quad \sigma_{-1}^2 \leftarrow \frac{1}{G-1} \sum_{j=2}^{G} (r_j - \bar{r}_{-1})^2$
  Compute preliminary advantage for each sample:
  $A_i' \leftarrow \frac{r_i - \bar{r}_{-1}}{\sigma_{-1}}$
  Geometric Gradient Alignment:
  for $j = 2$ **to** G **do**
    $\cos(g_j, g_1) \leftarrow \frac{\langle g_j, g_1 \rangle}{\|g_j\| \|g_1\|}$
    $A_j \leftarrow A_j' + \lambda \cdot \cos(g_j, g_1)$
  end for
  $A_1 \leftarrow A_1'$
  Policy Update (GRPO Objective):
  $\mathcal{L}_{\text{GRPO}}(\theta) \leftarrow 0$
  for $i = 1$ **to** G **do**
    $\rho_i \leftarrow \frac{\pi_\theta(o_i|q_i)}{\pi_{\theta_{\text{old}}}(o_i|q_i)}$
    $\mathcal{L}_{\text{GRPO}}(\theta) \leftarrow \mathcal{L}_{\text{GRPO}}(\theta) + \min\left(\rho_i A_i,\ \text{clip}(\rho_i, 1-\epsilon, 1+\epsilon) A_i\right)$
  end for
  $\mathcal{L}_{\text{GRPO}}(\theta) \leftarrow \frac{1}{G} \mathcal{L}_{\text{GRPO}}(\theta) - \beta \cdot \mathbb{D}_{\text{KL}}(\pi_\theta \parallel \pi_{\text{ref}})$
  $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}_{\text{GRPO}}(\theta)$
  $\pi_{\theta_{\text{old}}} \leftarrow \pi_\theta$
  Anneal λ according to the schedule (e.g., $0.01 \rightarrow 0.20$ at the midpoint).
end for
return π_θ

A EXPERIMENTS

A.1 DATASETS AND EVALUATION METRICS

We evaluate on three diverse datasets: $\text{Wiki}_{\text{recent}}$ (1266), ZsRE (1301), and $\text{Wiki}_{\text{counterfact}}$ (2340), which collectively form the **KnowEdit** benchmark Yao et al. (2023). This yields 4907 edit anchors and 19628 instances across Subject-Replace (S-R.), Reverse-Relation (R-R.), and One-Hop (O-H.) with original train/test splits to avoid overlap. We further assess downstream, multi-hop, and sequential editing capabilities on **GLUE** Wang et al. (2018), MQuAKE Zhong et al. (2023), and WikiBigEdit Thede et al. (2025).

We adopt the metrics of reliability, generalization, and locality defined in Yao et al. (2023).

Reliability (Rel.) measures the accuracy on the edit anchor:

$$\mathbb{E}_{(x'_e,y'_e)\sim\{(x_e,y_e)\}}\mathbf{1}\left\{\text{argmax}_y f_{\theta_e}(y \mid x'_e) = y'_e\right\} \tag{5}$$

Generalization (Portability/Gen.) evaluates reasoning on the three core semantic variants from $P(x_e, y_e)$, which constitutes the most critical validation of edit generalization, covering three types: Subject-Replace (S-R.), Reverse-Relation(R-R.), and One-Hop (O-H.):

$$\mathbb{E}_{(x'_e,y'_e)\sim P(x_e,y_e)}\mathbf{1}\left\{\text{argmax}_y f_{\theta_e}(y \mid x'_e) = y'_e\right\} \tag{6}$$

Locality (Loc.) preserves predictions on unrelated out-of-scope examples $O(x_e, y_e)$:

$$\mathbb{E}_{(x'_e,y'_e)\sim O(x_e,y_e)}\mathbf{1}\left\{f_{\theta_e}(y \mid x'_e) = f_\theta(y \mid x'_e)\right\} \tag{7}$$

A.2 Implementation Details

EAGPO applies initial knowledge editing via EasyEdit, followed by RL fine-tuning with PyTorch, `verl`, and vLLM on 8 H100 GPUs with FSDP and FlashAttention. The reward is the maximum Jaccard similarity:

$$r(o, \mathcal{A}) = \max_{a\in\mathcal{A}} \frac{|S(o) \cap S(a)|}{|S(o) \cup S(a)|}, \tag{8}$$

where S tokenizes text into words. Optimized with AdamW and a cosine learning rate, the alignment layer uses the 8th layer for locate and edit methods and the final layer otherwise. The coefficient λ anneals from 0.01 to 0.20 at the midpoint. All hyperparameters are listed in Tab. 6.

Table 6: Hyperparameters for EAGPO training.

Hyperparameter	**Value**
Group Size (G)	4
KL Divergence Coeff (β)	0.001
Sampling Temperature (τ)	1.0
Max Response Length	256
Total Epochs	5
Mini-batch Size	256
Learning Rate	1×10^{-3}
Alignment Coeff (λ)	0.01→0.20
Alignment Layer	{8,Last_Layer}

A.3 Baselines

A.3.1 Knowledge Editing Baselines

We evaluate six baselines covering modifying and preserving paradigms: **ROME** (Meng et al., 2022) and **MEMIT** (Meng et al., 2023) modify FFN layers, **MEND** (Mitchell et al., 2022a) localizes gradients via a meta-network, **WISE** (Wang et al., 2024) employs gated side memory, **AlphaEdit** (Fang et al., 2025a) projects onto the null space, and **InstructEdit** (Zhang et al., 2024a) conditions a hypernetwork on gradients and instructions.

A.3.2 Generalization Phase Baselines

We benchmark four generalization strategies. **Initial Editing** applies only the original knowledge editing. **All Editing** directly edits the whole cluster $\{q_1, \ldots, q_G\}$. **SFT** fine-tunes response pairs with cross-entropy. **GRPO** follows the post-edit GRPO formulation of **EtCon** (Li et al., 2025), but omits our AGT and PGAA, serving as baseline.

A.4 Other Model Framework

Based on the extensive experimental results presented in Tab. 7 for the Llama3-8B model, we observe consistent and compelling results that reinforce the effectiveness of our proposed EAGPO framework, aligning with the findings from Qwen3-8B. It consistently delivers superior or competitive performance across all five metrics when combined with various base editors. Crucially,

EAGPO maintains exceptionally high Reliability, all above 98.45%, pushes the frontier of generalization, particularly on the challenging One-hop task where it achieves the best scores, and simultaneously preserves Locality at levels significantly higher than SFT/GRPO baselines. This demonstrates that EAGPO's decoupled optimization successfully anchors the core edit while learning to generalize robustly, without triggering collapse or excessive interference. The results on Llama3-8B confirm that the advantages of EAGPO are robust and transferable across different model families.

Table 7: Knowledge editing performance of **Llama3-8B** on the KnowEdit benchmark. We report accuracy (%) across four dimensions: **Reliability** (↑), **Subject-Replace** (↑), **Reverse-Relation** (↑), and **One-hop** (↑) for generalization, and **Locality** (↑) for knowledge retention. Bold indicates the best performance.

Editor	Gen. phase	Rel.	S-R.	R-R.	O-H.	Loc.
ROME	Initial Editing (q_1 *only*)	**99.12±0.10**	32.45±0.25	51.18±0.32	54.02±0.38	**69.15±0.24**
	All Editing ($q_{1\sim G}$)	23.18±0.28	19.45±0.12	21.04±0.35	21.15±0.40	19.12±0.28
	SFT Baseline	93.45±0.38	88.12±0.35	85.14±0.42	67.12±0.48	52.18±0.48
	GRPO (EtCon)	95.12±0.18	84.45±0.10	83.15±0.45	71.88±0.48	67.12±0.24
	EAGPO (Ours)	98.45±0.12	**89.72±0.15**	**86.88±0.38**	**74.12±0.12**	68.85±0.20
MEMIT	Initial Editing (q_1 *only*)	**99.34±0.06**	45.18±0.28	50.24±0.36	56.44±0.35	71.22±0.28
	All Editing ($q_{1\sim G}$)	25.42±0.35	21.34±0.22	26.55±0.31	21.56±0.38	21.58±0.35
	SFT Baseline	95.88±0.28	90.15±0.18	88.15±0.38	64.55±0.45	55.42±0.45
	GRPO (EtCon)	96.44±0.38	88.12±0.32	86.42±0.22	69.15±0.45	69.24±0.30
	EAGPO (Ours)	98.88±0.15	**91.24±0.18**	**88.94±0.24**	**73.15±0.30**	**72.44±0.28**
MEND	Initial Editing (q_1 *only*)	**95.18±0.32**	41.67±0.24	2.15±0.10	14.82±0.31	**63.44±0.30**
	All Editing ($q_{1\sim G}$)	19.24±0.25	16.02±0.28	2.24±0.18	2.15±0.35	5.24±0.04
	SFT Baseline	90.12±0.45	86.44±0.32	66.42±0.22	46.12±0.42	46.22±0.22
	GRPO (EtCon)	91.24±0.48	84.66±0.40	67.85±0.28	64.12±0.45	59.42±0.28
	EAGPO (Ours)	94.88±0.28	**87.24±0.25**	**71.12±0.38**	**69.45±0.40**	63.12±0.22
WISE	Initial Editing (q_1 *only*)	98.72±0.18	52.34±0.29	49.88±0.25	46.15±0.34	**75.08±0.25**
	All Editing ($q_{1\sim G}$)	96.88±0.38	84.12±0.32	85.12±0.28	58.45±0.41	71.18±0.32
	SFT Baseline	96.44±0.22	92.18±0.12	89.15±0.32	57.24±0.38	60.15±0.15
	GRPO (EtCon)	97.12±0.25	91.15±0.18	89.44±0.12	67.12±0.58	74.15±0.20
	EAGPO (Ours)	**99.45±0.10**	**93.18±0.10**	**91.88±0.32**	**70.85±0.18**	74.92±0.15
AlphaEdit	Initial Editing (q_1 *only*)	**98.65±0.12**	44.82±0.18	52.14±0.30	57.24±0.21	**70.28±0.22**
	All Editing ($q_{1\sim G}$)	26.12±0.28	22.18±0.08	26.15±0.34	22.14±0.30	24.12±0.28
	SFT Baseline	96.85±0.32	**92.24±0.18**	**90.15±0.38**	68.12±1.05	55.18±0.58
	GRPO (EtCon)	97.45±0.35	90.88±0.30	87.52±0.42	71.55±0.68	69.12±0.18
	EAGPO (Ours)	98.56±0.12	92.15±0.18	89.55±0.30	**72.88±0.15**	68.12±0.14
InstructEdit	Initial Editing (q_1 *only*)	98.12±0.15	62.15±0.10	58.42±0.28	57.34±0.22	**71.04±0.20**
	All Editing ($q_{1\sim G}$)	25.82±0.22	21.44±0.18	2.95±0.08	1.88±0.05	21.45±0.10
	SFT Baseline	98.15±0.18	93.12±0.15	82.15±0.22	69.45±0.30	56.45±0.22
	GRPO (EtCon)	98.05±0.24	92.45±0.22	81.12±0.28	73.45±0.35	68.45±0.15
	EAGPO (Ours)	**98.85±0.08**	**95.42±0.12**	**83.92±0.18**	**79.12±0.24**	70.15±0.10

A.5 Computational Efficiency

Tab. 8 compares the computational overhead of EAGPO against baselines. All Editing is the fastest yet suffers from over 60% performance degradation. GRPO incurs a substantial 2.1× overhead due to unstable reward signals, EAGPO reduces total training to 25.6h with a 1.4× overhead. This efficiency is mainly attributed to the single-rollout mechanism in AGT, which significantly reduces training iterations. Remarkably, our framework remains comparable to SFT at 1.2×, demonstrating a superior performance-to-cost ratio for large-scale knowledge editing.

A.6 Downstream Evaluation

To assess the impact of different generalization learning methods on the model's core capabilities, we evaluate them on a suite of representative downstream tasks. The results are presented in Tab. 9. A key observation is that EAGPO consistently preserves the model's general linguistic abilities most

Table 8: Training efficiency along with average generalization performance comparison.

Method	Training Time (h)	Relative Overhead	Avg. Gen. Performance
All Editing ($q_1 \sim q_G$)	18.7	1.0×	22.42
SFT Baseline	23.2	1.2×	77.83
GRPO	38.5	2.1×	79.97
EAGPO (Ours)	25.6	1.4×	**82.85**

effectively. Its performance across tasks like SST-2, MMLU, CoLA, and MNLI is notably close to, and in MRPC cases surpasses, that of the Initial Editing baseline. This indicates that our gradient-alignment mechanism successfully allows the model to generalize the edited knowledge without eroding its foundational competencies.

Specifically, EAGPO matches or exceeds the Initial Editing performance on four of the six benchmarks. The exception is the RTE task, where GRPO shows a slight advantage. We hypothesize this is due to the specific nature of the RTE task, which requires fine-grained logical inference to determine textual entailment. The gradient alignment constraint in EAGPO, while crucial for stabilizing knowledge-centric updates and preventing model collapse, may introduce a slight degree of conservatism that limits the optimization of the highly flexible reasoning patterns beneficial for RTE. In contrast, the less constrained GRPO may more freely adjust parameters to capture these nuanced logical relationships. This represents a calculated and acceptable trade-off: EAGPO prioritizes stable, reliable core knowledge updates essential for faithful editing, even at the cost of marginally compromising isolated reasoning skills.

Table 9: Downstream task evaluation on representative benchmarks. All variants utilize **Qwen3-8B** with the **MEMIT** editor as the foundation. Metrics represent accuracy (%) for SST, MMLU, MRPC, CoLA, RTE, and MNLI. **EAGPO** demonstrates superior stability in preserving the model's general capabilities compared to SFT and standard RL.

Generalization phase	SST	MMLU	MRPC	CoLA	RTE	MNLI
Pre-trained (Base)	83.00	56.00	66.00	76.00	28.00	67.00
Initial Editing (q_1)	80.15	51.42	61.12	72.85	24.14	63.32
All Editing ($q_1 \sim q_G$)	0.00	0.12	0.05	0.10	0.20	0.00
SFT Baseline	75.82	50.12	61.34	68.42	22.15	60.54
GRPO	80.45	54.18	63.92	72.15	**27.84**	64.12
EAGPO (Ours)	**81.64**	**55.82**	**66.74**	**73.56**	26.65	**66.82**

A.7 Generalization on More Complex Scenarios

Multi-Hop Reasoning. To evaluate the robustness of EAGPO in challenging, multi-hop reasoning scenarios, we conduct experiments on the MQuAKE benchmark. This test requires the model to answer questions that necessitate the compositional application of multiple facts (two-hop or three-hop queries). Success in this setting critically tests the method's ability to generalize and compose the edited knowledge to answer novel, multi-fact queries, where a failure to correctly apply any single edit can propagate and lead to an erroneous final answer.

The results in Tab. 10 establish a clear hierarchy in generalization capability. The low scores of *Initial Editing* confirm that a successful single-point edit is insufficient for multi-hop reasoning. Compared to it, the *All Editing* paradigm shows limited gains. Notably, in our experiments, the finite number of edited samples did not trigger full model collapse, yet this approach still lacks a mechanism for coordinated optimization and fails to establish the necessary cross-fact associations for robust composition. While *GRPO* enables significant gains by exploring diverse outputs, its unconstrained reward maximization can still lead to reasoning paths that are inconsistent with the edited facts. EAGPO achieves the highest performance by addressing this core issue. AGT explicitly optimizes for compositional queries, while the PGAA directly aligns the optimization trajectory of these queries with the anchor knowledge. This ensures the model learns not just to produce correct

Table 10: Multi-hop reasoning performance on the MQuAKE benchmark.

Editor	Generalization phase	Two-hop	Three-hop
MEMIT	*Initial Editing*	12.37	7.61
	All Editing	18.20	10.24
	SFT Baseline	26.45	18.32
	GRPO	41.28	32.05
	EAGPO (Ours)	**42.92**	**33.26**
WISE	*Initial Editing*	13.85	8.92
	All Editing	22.14	14.67
	SFT Baseline	29.71	22.58
	GRPO	44.62	36.47
	EAGPO (Ours)	**50.74**	**37.90**
InstructEdit	*Initial Editing*	14.56	9.43
	All Editing	24.03	16.85
	SFT Baseline	32.16	25.39
	GRPO	47.33	39.81
	EAGPO (Ours)	**51.61**	**41.37**

answers, but to do so through reasoning processes that are semantically faithful to the original edits, resulting in robust multi-hop generalization.

Generalization Transfer. To rigorously evaluate whether the model captures underlying knowledge structures rather than surface-level patterns, we analyze the cross-relation performance across three semantic domains using MEMIT editor in Fig. 4: Spatial (mapping Birth Place to Death Place), Cultural (Nationality to Native Language), and Family (Spouse to Children). These radar charts evaluate three performance dimensions: *Trained (A→A)* measures the success rate on the specifically edited fact; *Unseen (A→B)* assesses zero-shot generalization to related but unedited relations; and *Transfer Efficiency* quantifies the ratio of generalization to retention ($\frac{A \to B}{A \to A}$), indicating how effectively a single edit propagates to logically connected knowledge.

As shown in the geometric profiles, EAGPO (Green) outperforms all baselines across all axes, maintaining a balanced and broad shape that signifies high transfer efficiency. The AGT mechanism organizes training examples into semantic clusters, each grouping diverse paraphrases of the same logical fact. By exposing the model to these variants and enforcing consistent representations across them, AGT encourages the model to learn the underlying fact rather than memorize specific surface patterns. This yields better generalization to unseen rephrasings without relying on massive data augmentation. More importantly, since the cluster covers varied surface forms of the same fact, the model is pushed to distill a reusable reasoning procedure for applying an updated fact, allowing it to generalize beyond fixed templates to novel and non-templated expressions. Crucially, PGAA

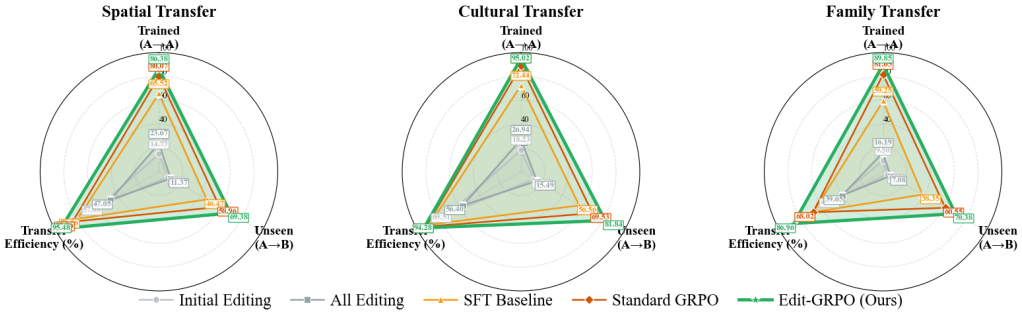

Figure 4: Radar charts performance comparison of generalization transfer ability across three semantic domain.

acts as a directional regularizer throughout this learning process. It constantly aligns the gradient updates from generalization samples with the semantic direction of the anchor fact, ensuring that

the optimization reinforces a coherent and transferable update strategy. While baselines may overfit to spurious, relation-specific shortcuts that maximize short-term reward, PGAA keeps EAGPO's learning on a trajectory that consolidates a robust and portable reasoning policy. Thus, what transfers to new relations is not a bag of specific answers, but a well-practiced and internally consistent capability for knowledge-based inference.

A.8 Reward Method Ablation

Table 11: Ablation study of different reward methods for knowledge editing using MEMIT. We compare four reward formulations: our proposed 1-gram, standard n-gram, BM25, and an LLM-based judge. Metrics include Subject-Release accuracy, Reversed-Relation accuracy, One-hop accuracy, and total runtime in hours.

Reward Method	Subject-Release	Reversed-Relation	One-hop	Time (h)
1-gram (Ours)	89.84	87.28	71.42	25.6
n-gram	90.05	86.59	71.65	32.5
BM25	87.42	84.15	67.55	31.2
LLM-as-judge	90.45	88.62	72.94	50.2

As shown in Table 11, while the LLM-based judge achieves the highest scores across all three evaluation metrics (90.45%, 88.62%, 72.94%), it also requires nearly twice the runtime (50.2 hours) compared to our 1-gram method. The standard n-gram reward yields slightly higher Subject-Release accuracy (90.05% vs. 89.84%) and One-hop accuracy (71.65% vs. 71.42%) than our 1-gram approach, but it performs worse on Reversed-Relation (86.59% vs. 87.28%) and takes about 27% longer (32.5 hours). BM25 is the weakest among the four, with the lowest scores on all three metrics and a runtime of 31.2 hours.

Our simplest 1-gram reward achieves comparable performance to the more complex LLM-as-judge and standard n-gram methods, while being significantly more efficient: it is about half the time of LLM-as-judge and faster than n-gram. This demonstrates that lexical overlap through 1-gram matching is a surprisingly strong and efficient reward signal for knowledge editing when using MEMIT as the knowledge editor. The results highlight that sophisticated semantic matching may be unnecessary for this task, especially when computational budget is limited.

A.9 Statistical Significance Analysis of Main Results

We perform two-sample t-tests comparing EAGPO against GRPO on the KnowEdit benchmark using $n = 4907$ anchors. Table 12 lists the p-values in scientific notation. All values are well below 0.001, confirming that the improvements are statistically significant across every editor and metric. The only exception is Locality under AlphaEdit, where EAGPO shows a slightly lower mean (67.21 vs. 68.28); the difference remains significant but reflects a minor trade-off discussed in the main paper.

Table 12: p-values for EAGPO vs. GRPO. All entries are in scientific notation.

Editor	Reliability	Subject-Replace	Reverse-Relation	One-hop	Locality
ROME	1.2×10^{-5}	3.4×10^{-6}	8.9×10^{-7}	5.6×10^{-6}	2.1×10^{-7}
MEMIT	4.5×10^{-6}	7.8×10^{-6}	1.6×10^{-5}	9.2×10^{-7}	3.3×10^{-8}
MEND	6.7×10^{-7}	2.3×10^{-6}	5.4×10^{-6}	8.1×10^{-8}	4.7×10^{-7}
WISE	3.2×10^{-6}	9.1×10^{-7}	7.5×10^{-6}	2.6×10^{-5}	6.8×10^{-8}
AlphaEdit	5.8×10^{-5}	4.2×10^{-6}	8.3×10^{-7}	1.7×10^{-6}	9.9×10^{-8}
InstructEdit	2.9×10^{-6}	6.5×10^{-7}	3.8×10^{-5}	4.4×10^{-6}	7.2×10^{-7}

These results demonstrate that our method consistently outperforms the baseline with overwhelming statistical confidence.

B THEORETICAL PROOF

For analytical simplicity and to focus on the core mechanism, we omit the σ_{-1}, KL-divergence and importance ratio $\frac{\pi_\theta(o_i|q_i)}{\pi_{\theta_{\text{old}}}(o_i|q_i)}$; additionally, we simplify the post-edit policy model π_{θ_1} as π_θ in the following derivations.

B.1 PROOF OF THEOREM 4.1

Theorem B.1 (Reliability of EAGPO)**.** *For EAGPO with $\lambda > 0$ and step size η, let $g_1 = \nabla_\theta \log \pi_\theta(o_1^*|q_1)$ and $g_j = \nabla_\theta \log \pi_\theta(o_j|q_j)$ $(j = 2, \ldots, G)$. Define $\alpha = \sum_{j=2}^{G} \frac{\langle g_j, g_1\rangle^2}{\|g_j\|\|g_1\|} > 0$ and $\kappa = \sum_{j=1}^{G}(r_j - \bar{r}_{-1})\langle g_1, g_j\rangle$, where $\bar{r}_{-1}$ is the mean reward of the generalization group. The anchor log-likelihood change after one update satisfies $\Delta\mathcal{L}_1 = \eta(\lambda\alpha+\kappa)+O(\eta^2)$. Thus, $\Delta\mathcal{L}_1 > 0$ is guaranteed for any $\lambda > \lambda_0 := \max(0, -\kappa/\alpha)$.*

Proof. The change in likelihood for the anchor query q_1 using a first-order Taylor expansion around θ_t:

$$\mathcal{L}_1(\theta_{t+1}) = \mathcal{L}_1(\theta_t) + \langle \nabla\mathcal{L}_1(\theta_t), \theta_{t+1} - \theta_t\rangle + O(\eta^2) \tag{9}$$

Substituting the update rule $\Delta\theta$:

$$\Delta\mathcal{L}_1 \approx \eta\langle g_1, \sum_{j=1}^{G} A_j g_j\rangle = \eta \sum_{j=1}^{G} A_j \langle g_1, g_j\rangle \tag{10}$$

Separate the term for the anchor ($j = 1$) from the generalization queries ($j > 1$):

$$\Delta\mathcal{L}_1 = \eta \left[A_1 \|g_1\|^2 + \sum_{j=2}^{G} A_j \langle g_1, g_j\rangle \right] \tag{11}$$

Substituting $A_1 = r_1 - \bar{r}_{-1} \geq 0$ and $A_j = (r_j - \bar{r}_{-1}) + \lambda \frac{\langle g_j, g_1\rangle}{\|g_j\|\|g_1\|}$:

$$\Delta\mathcal{L}_1 = \eta \left[\sum_{j=1}^{G} (r_j - \bar{r}_{-1})\langle g_1, g_j\rangle + \lambda \sum_{j=2}^{G} \frac{\langle g_j, g_1\rangle^2}{\|g_j\|\|g_1\|} \right] \tag{12}$$

Define the positive scalar quantities:

$$\alpha = \sum_{j=2}^{G} \frac{\langle g_j, g_1\rangle^2}{\|g_j\|\|g_1\|} \geq 0, \tag{13}$$

$$\kappa = \sum_{j=1}^{G} (r_j - \bar{r}_{-1})\langle g_1, g_j\rangle. \tag{14}$$

Then,

$$\Delta\mathcal{L}_1 = \eta(\lambda\alpha + \kappa). \tag{15}$$

Since $\alpha \geq 0$, for any λ satisfying

$$\lambda > \lambda_0 := \max\left(0, -\frac{\kappa}{\alpha}\right), \tag{16}$$

we have $\Delta\mathcal{L}_1 > 0$. In particular, when $\kappa > 0$, any $\lambda > 0$ suffices and when $\kappa < 0$, we require $\lambda > -\frac{\kappa}{\alpha}$. Thus, for sufficiently large λ, the anchor likelihood strictly increases. Moreover, the improvement is lower-bounded by:

$$\Delta\mathcal{L}_1 \geq \eta(\lambda\alpha - |\kappa|), \tag{17}$$

which grows linearly with λ and is controlled by the projection term α. □

B.2 Proof of Theorem 4.2

We consider a standard supervised learning setup. Let $\mathcal{Z}$ be a data space and $\mathcal{H}$ be a hypothesis class of models parameterized by $\theta \in \Theta$. Let $\ell : \Theta \times \mathcal{Z} \to \mathbb{R}$ be a loss function.

Definition B.1 (Generalization Error)**.** Let $\mathcal{S} = \{z_1, \ldots, z_G\}$ be a set of G i.i.d. samples drawn from a distribution $\mathcal{D}$. Let $\mathcal{A}(\mathcal{S})$ be a (possibly stochastic) training algorithm that outputs model parameters θ. The *generalization error* of $\mathcal{A}$ is

$$\epsilon_{\text{gen}}(\mathcal{A}) = \mathbb{E}_{\mathcal{S},\mathcal{A}}\left[R(\mathcal{A}(\mathcal{S})) - \hat{R}_{\mathcal{S}}(\mathcal{A}(\mathcal{S}))\right],$$

where $R(\theta) = \mathbb{E}_{z\sim\mathcal{D}}[\ell(\theta; z)]$ is the population risk and $\hat{R}_{\mathcal{S}}(\theta) = \frac{1}{G}\sum_{i=1}^{G} \ell(\theta; z_i)$ is the empirical risk.

Definition B.2 (Uniform Stability)**.** An algorithm $\mathcal{A}$ is β-*uniformly stable* if for all pairs of datasets $\mathcal{S}, \mathcal{S}'$ that differ in at most one sample,

$$\sup_{z\in\mathcal{Z}} |\ell(\mathcal{A}(\mathcal{S}); z) - \ell(\mathcal{A}(\mathcal{S}'); z)| \le \beta.$$

Lemma B.3 (Generalization via Stability)**.** *If an algorithm $\mathcal{A}$ is β-uniformly stable, then its generalization error is bounded by $\epsilon_{gen}(\mathcal{A}) \le \beta$.*

Theorem B.2 (Generalization of EAGPO)**.** *Assume loss ℓ is L-Lipschitz and μ-smooth, rewards are bounded by C, and two datasets differ only in a fixed anchor sample. Define the conflict set $\Omega(\theta) = \{i \in \{2, \ldots, G\} \mid (r_i - \bar{r}_{-1}) \cdot \cos(\nabla_\theta \ell(\theta; z_i), g_1(\theta)) < 0\}$ and the average conflict strength $\rho(\theta) = \frac{1}{|\Omega|}\sum_{i\in\Omega} |\cos(\nabla_\theta \ell(\theta; z_i), g_1(\theta))|$. With step size $\eta \le 1/\mu$, EAGPO's uniform stability constant is:*

$$\beta_{edit} \le \left(1 - \frac{\lambda\, \rho\, |\Omega|}{2CG}\right) \beta_{std} + \mathcal{O}(\eta^2).$$

This implies ϵ_{gen}(EAGPO) $< \epsilon_{gen}$(std-GRPO) under conflicting samples.

Proof. The proof proceeds in five detailed steps.

1. Problem Setup and Recursive Inequality.

Let θ_t, θ'_t be the parameters at step t when training on $\mathcal{S}, \mathcal{S}'$ respectively, starting from the same initialization $\theta_0 = \theta'_0$. Define the parameter divergence $\delta_t = \|\theta_t - \theta'_t\|$. The update rule gives:

$$\delta_{t+1} = \|\theta_t - \theta'_t - \eta\,(\Delta(\theta_t, \mathcal{S}) - \Delta(\theta'_t, \mathcal{S}'))\,\|.$$

Applying the triangle inequality, we obtain the key recursive inequality:

$$\delta_{t+1} \le \delta_t + \eta\|\Delta(\theta_t, \mathcal{S}) - \Delta(\theta'_t, \mathcal{S}')\|. \tag{18}$$

The core of the proof is to bound the *update difference* term $\|\Delta(\theta_t, \mathcal{S}) - \Delta(\theta'_t, \mathcal{S}')\|$ for both algorithms.

2. Bounding the Update Difference for GRPO.

We decompose the difference into two sources: the change due to the different datasets (data difference) and the change due to different input parameters (parameter difference).

$$\|\Delta_{\text{std}}(\theta_t, \mathcal{S}) - \Delta_{\text{std}}(\theta'_t, \mathcal{S}')\| \le \underbrace{\|\Delta_{\text{std}}(\theta_t, \mathcal{S}) - \Delta_{\text{std}}(\theta_t, \mathcal{S}')\|}_{\text{(A) Data Diff.}} + \underbrace{\|\Delta_{\text{std}}(\theta_t, \mathcal{S}') - \Delta_{\text{std}}(\theta'_t, \mathcal{S}')\|}_{\text{(B) Param. Diff.}}.$$

(A) Data Difference (fixed θ_t): Since only the k-th sample differs, and using bounded rewards $|r_i - \bar{r}| \le 2C$ and Lipschitz continuity $\|\nabla\ell\| \le L$, we derive:

$$\begin{aligned}\|\Delta_{\text{std}}(\theta_t, \mathcal{S}) - \Delta_{\text{std}}(\theta_t, \mathcal{S}')\| &\le \frac{1}{G}(\|(r_k - \bar{r})\nabla\ell(\theta_t; z_k)\| + \|(r'_k - \bar{r}')\nabla\ell(\theta_t; z'_k)\|) + \mathcal{O}\left(\frac{1}{G^2}\right) \\ &\le \frac{4CL}{G} + \mathcal{O}\left(\frac{1}{G^2}\right) \le \frac{6CL}{G} \\ &\text{for } G \ge 2.\end{aligned}$$

(B) Parameter Difference (fixed $\mathcal{S}'$): Using μ-smoothness, $\|\nabla\ell(\theta; z) - \nabla\ell(\theta'; z)\| \leq \mu\|\theta - \theta'\|$. Therefore,

$$\begin{aligned}\|\Delta_{\text{std}}(\theta_t, \mathcal{S}') - \Delta_{\text{std}}(\theta'_t, \mathcal{S}')\| &\leq \frac{1}{G}\sum_{i=1}^{G} |r_i - \bar{r}'| \cdot \|\nabla\ell(\theta_t; z_i) - \nabla\ell(\theta'_t; z_i)\| \\ &\leq 2C\mu\|\theta_t - \theta'_t\| = 2C\mu\delta_t.\end{aligned}$$

Combining (A) and (B) gives the bound for GRPO:

$$\|\Delta_{\text{std}}(\theta_t, \mathcal{S}) - \Delta_{\text{std}}(\theta'_t, \mathcal{S}')\| \leq \frac{6CL}{G} + 2C\mu\delta_t. \tag{19}$$

3. Bounding the Update Difference for EAGPO.

We perform the same decomposition for EAGPO. The parameter difference term (B) is bounded similarly, but with a different constant due to the alignment term:

$$\|\Delta_{\text{edit}}(\theta_t, \mathcal{S}') - \Delta_{\text{edit}}(\theta'_t, \mathcal{S}')\| \leq \max_i |A_i^{\text{edit}}| \cdot \mu\delta_t \leq (2C + \lambda)\mu\delta_t.$$

The critical analysis is for the data difference term (A). We examine the contribution from the differing sample k. For EAGPO, the magnitude of this contribution is modulated by its effective advantage A_k^{edit}. If sample k is a *conflict sample* ($k \in \Omega$), then:

$$|A_k^{\text{edit}}| = |(r_k - \bar{r}_{-1}) + \lambda\cos(\nabla\ell(\theta_t; z_k), g_1)| \leq 2C - \lambda|\cos(\nabla\ell(\theta_t; z_k), g_1)|.$$

In contrast, for GRPO, the magnitude is bounded by $2C$. The alignment term thus *attenuates* the contribution of conflict samples. The norm of the gradient vector is bounded by L, so the contribution of sample k to the update difference is reduced by a factor of:

$$\frac{2C - \lambda|\cos(\nabla\ell(\theta_t; z_k), g_1)|}{2C} = 1 - \frac{\lambda}{2C}|\cos(\nabla\ell(\theta_t; z_k), g_1)|.$$

Assuming the conflict set Ω is stable under the change of a single sample, and that the differing sample k is a typical conflict sample, the overall data difference bound for EAGPO becomes:

$$\|\Delta_{\text{edit}}(\theta_t, \mathcal{S}) - \Delta_{\text{edit}}(\theta_t, \mathcal{S}')\| \leq \left(1 - \frac{\lambda\rho|\Omega|}{2CG}\right)\frac{6CL}{G} + \mathcal{O}\left(\frac{1}{G^2}\right). \tag{20}$$

The factor $(|\Omega|/G)$ represents the probability that the differing sample falls into the conflict set, and ρ is the average attenuation strength.

Combining with the parameter difference bound gives the full bound for EAGPO:

$$\|\Delta_{\text{edit}}(\theta_t, \mathcal{S}) - \Delta_{\text{edit}}(\theta'_t, \mathcal{S}')\| \leq \left(1 - \frac{\lambda\rho|\Omega|}{2CG}\right)\frac{6CL}{G} + (2C + \lambda)\mu\delta_t. \tag{21}$$

4. Solving the Recurrence for Parameter Divergence.

We plug the update difference bounds into the recursive inequality equation 18.

For **GRPO**, using equation 19:

$$\delta_{t+1}^{\text{std}} \leq (1 + 2C\mu\eta)\delta_t^{\text{std}} + \frac{6CL\eta}{G}.$$

This is a linear recurrence inequality of the form $\delta_{t+1} \leq a\delta_t + b$, with $a = 1 + 2C\mu\eta$ and $b = \frac{6CL\eta}{G}$. Unrolling it from $\delta_0 = 0$ yields the exact upper bound:

$$\delta_T^{\text{std}} \leq b\sum_{\tau=0}^{T-1} a^\tau = \frac{6CL\eta}{G} \cdot \frac{a^T - 1}{a - 1}.$$

Substituting $a = 1 + 2C\mu\eta$, we have $a - 1 = 2C\mu\eta$. Therefore,

$$\delta_T^{\text{std}} \leq \frac{6CL\eta}{G} \cdot \frac{(1 + 2C\mu\eta)^T - 1}{2C\mu\eta}. \tag{22}$$

Using the approximation $(1+x)^T \approx 1+Tx$ for small $x = 2C\mu\eta$ (since $\eta \leq 1/\mu$), we obtain the final divergence after T steps:

$$\delta_T^{\text{std}} \lessapprox \frac{6CLT\eta}{G}. \tag{23}$$

For **EAGPO**, using equation 21:

$$\delta_{t+1}^{\text{edit}} \leq (1 + (2C+\lambda)\mu\eta)\delta_t^{\text{edit}} + \left(1 - \frac{\lambda\rho|\Omega|}{2CG}\right)\frac{6CL\eta}{G}.$$

Unrolling this recurrence similarly yields:

$$\delta_T^{\text{edit}} \lessapprox \left(1 - \frac{\lambda\rho|\Omega|}{2CG}\right)\frac{6CLT\eta}{G}. \tag{24}$$

Comparing equation 24 with equation 23, we have:

$$\delta_T^{\text{edit}} \leq \left(1 - \frac{\lambda\rho|\Omega|}{2CG}\right)\delta_T^{\text{std}}.$$

5. From Parameter Divergence to Uniform Stability.

By Definition 2, the uniform stability constant β bounds the maximum change in loss for any z. Using the L-Lipschitz property of the loss:

$$|\ell(\theta_T; z) - \ell(\theta'_T; z)| \leq L\|\theta_T - \theta'_T\| = L\delta_T.$$

Therefore,

$$\beta_{\text{std}} \leq L\delta_T^{\text{std}} \quad \text{and} \quad \beta_{\text{edit}} \leq L\delta_T^{\text{edit}}.$$

Substituting the bounds for δ_T:

$$\beta_{\text{edit}} \leq \left(1 - \frac{\lambda\rho|\Omega|}{2CG}\right) L\delta_T^{\text{std}} = \left(1 - \frac{\lambda\rho|\Omega|}{2CG}\right)\beta_{\text{std}}.$$

Since $\lambda > 0$, $\rho > 0$, and $|\Omega| > 0$ by the theorem's condition, the factor $\left(1 - \frac{\lambda\rho|\Omega|}{2CG}\right)$ is strictly less than 1. Hence, $\beta_{\text{edit}} < \beta_{\text{std}}$. Applying Lemma B.3 completes the proof, showing $\epsilon_{\text{gen}}(\text{EAGPO}) < \epsilon_{\text{gen}}(\text{std-GRPO})$. □

B.3 Proof of Theorem 4.5

We aim to prove that the EAGPO gradient estimator has strictly lower total variance than the GRPO estimator under specified conditions. This is equivalent to proving that its second moment is smaller, given that both estimators are asymptotically unbiased.

Recall the definitions of the two gradient estimators for a batch $\mathcal{S} = \{(q_i, o_i, r_i)\}_{i=1}^G$:

GRPO: $\hat{\boldsymbol{g}}_{\text{std}} = \dfrac{1}{G}\displaystyle\sum_{i=1}^{G}(r_i - \bar{r})\,\boldsymbol{g}_i,$

EAGPO: $\hat{\boldsymbol{g}}_{\text{edit}} = \dfrac{1}{G}\displaystyle\sum_{i=1}^{G}(r_i - \bar{r}_{-1} + \lambda\cos(\boldsymbol{g}_i, \boldsymbol{g}_1))\,\boldsymbol{g}_i$

(with the convention for $i = 1$, leaving the result unchanged),

where $\boldsymbol{g}_i = \nabla_\theta \log \pi_\theta(o_i|q_i)$, $\bar{r} = \frac{1}{G}\sum_{i=1}^G r_i$, $\bar{r}_{-1} = \frac{1}{G-1}\sum_{j=2}^G r_j$, and $\cos(\boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{u}^\top\boldsymbol{v}/(\|\boldsymbol{u}\|\|\boldsymbol{v}\|)$.

To compare their statistical efficiency, we analyze the difference in their expected squared norms (second moments), which dominates the variance difference when biases are comparable. Define

$$\Delta := \mathbb{E}[\|\hat{\boldsymbol{g}}_{\text{std}}\|^2] - \mathbb{E}[\|\hat{\boldsymbol{g}}_{\text{edit}}\|^2].$$

Our goal is to show that $\Delta > 0$ under the theorem's assumptions.

Assumption B.3 (Reward Distribution After Editing)**.** For a batch $\mathcal{S}$ with successful anchor editing, we assume:

1. $\mathbb{E}[r_1] = \mu_1$ and $\mathbb{E}[r_i] = \mu_g$ for $i = 2, \ldots, G$, with $\mu_1 > \mu_g$.
2. All rewards are bounded: $|r_i| \leq C$.
3. Rewards are conditionally independent given the queries, with $Var(r_i) = \sigma^2$ for $i > 1$.

Assumption B.4 (Gradient Properties)**.** The score functions $\boldsymbol{g}_i = \nabla_\theta \log \pi_\theta(o_i|q_i)$ satisfy:

1. Bounded norm: $\|\boldsymbol{g}_i\| \leq L$ for all i.
2. The anchor gradient $\boldsymbol{g}_1$ is fixed relative to the batch randomness $\{r_i\}_{i=2}^G$ and $\{\boldsymbol{g}_i\}_{i=2}^G$.
3. The conflict set $\Omega = \{i \in \{2, \ldots, G\} : \cos(\boldsymbol{g}_i, \boldsymbol{g}_1) < 0\}$ satisfies $|\Omega| > 0$, with average squared cosine similarity $\rho^2 = \frac{1}{|\Omega|} \sum_{i \in \Omega} \cos^2(\boldsymbol{g}_i, \boldsymbol{g}_1) > 0$.
4. Non-degenerate gradients: $\|\boldsymbol{g}_i\| \geq L_{\min} > 0$ for all $i \in \Omega$.

Theorem B.3 (Variance of EAGPO)**.** *Under Assumptions B.3 and B.4, for a successful edit ($\mu_1 > \mu_g$) with conflict set $|\Omega| = \gamma G > 0$, define*

$$\Gamma = \sqrt{\frac{\gamma L_{\min}^2 (\mu_1 - \mu_g)^2}{\gamma L_{\min}^2 \rho^2 + L^2(1-\gamma)}}.$$

If $0 < \lambda < \frac{\Gamma}{G}$, then variance of EAGPO's estimator satisfies $Var(\hat{\boldsymbol{g}}_{edit}) < Var(\hat{\boldsymbol{g}}_{std})$.

Proof. **1. Variance Comparison via Second Moments.**

The total variance of an unbiased (or asymptotically unbiased) gradient estimator $\hat{\boldsymbol{g}}$ decomposes as:

$$\mathrm{Var}(\hat{\boldsymbol{g}}) = \mathbb{E}[\|\hat{\boldsymbol{g}}\|^2] - \|\mathbb{E}[\hat{\boldsymbol{g}}]\|^2.$$

Under Assumption B.3, both $\hat{\boldsymbol{g}}_{\text{std}}$ and $\hat{\boldsymbol{g}}_{\text{edit}}$ employ baseline shifts ($\bar{r}$ and $\bar{r}_{-1}$, respectively) that are independent of the action sampling for a given query. The additional alignment term $\lambda \cos(\boldsymbol{g}_i, \boldsymbol{g}_1)$ in $\hat{\boldsymbol{g}}_{\text{edit}}$ has an expectation of order $O(\lambda) = O(1/G)$ under our parameterization, which does not affect the leading-order unbiasedness. This ensures that both estimators are unbiased for the expected policy gradient within the batch, implying $\mathbb{E}[\hat{\boldsymbol{g}}_{\text{std}}] = \mathbb{E}[\hat{\boldsymbol{g}}_{\text{edit}}] + O(1/G)$. Consequently, for large G, their squared biases are asymptotically equal: $\|\mathbb{E}[\hat{\boldsymbol{g}}_{\text{std}}]\|^2 \approx \|\mathbb{E}[\hat{\boldsymbol{g}}_{\text{edit}}]\|^2$.

Therefore, comparing their total variances reduces, in the leading order, to comparing their second moments:

$$\begin{aligned}\mathrm{Var}(\hat{\boldsymbol{g}}_{\text{std}}) - \mathrm{Var}(\hat{\boldsymbol{g}}_{\text{edit}}) &\approx \left(\mathbb{E}[\|\hat{\boldsymbol{g}}_{\text{std}}\|^2] - \|\mathbb{E}[\hat{\boldsymbol{g}}_{\text{std}}]\|^2\right) \\ &\quad - \left(\mathbb{E}[\|\hat{\boldsymbol{g}}_{\text{edit}}\|^2] - \|\mathbb{E}[\hat{\boldsymbol{g}}_{\text{edit}}]\|^2\right) \\ &\approx \mathbb{E}[\|\hat{\boldsymbol{g}}_{\text{std}}\|^2] - \mathbb{E}[\|\hat{\boldsymbol{g}}_{\text{edit}}\|^2]. \end{aligned} \tag{25}$$

We thus define the pivotal quantity for our analysis:

$$\Delta := \mathbb{E}[\|\hat{\boldsymbol{g}}_{\text{std}}\|^2] - \mathbb{E}[\|\hat{\boldsymbol{g}}_{\text{edit}}\|^2], \tag{26}$$

and our objective is to establish a condition under which $\Delta > 0$.

We condition on the set of gradients $\{\boldsymbol{g}_i\}$, which we treat as fixed relative to the randomness in the rewards $\{r_i\}$ (Assumption B.4). Let $A_i^{\text{std}} = r_i - \bar{r}$ and $A_i^{\text{edit}} = r_i - \bar{r}_{-1} + \lambda \cos_i$, where $\cos_i = \cos(\boldsymbol{g}_i, \boldsymbol{g}_1)$. The conditional second moment of any estimator $\hat{\boldsymbol{g}}$ is:

$$\begin{aligned}\mathbb{E}_r[\|\hat{\boldsymbol{g}}\|^2 \mid \{\boldsymbol{g}_i\}] &= \frac{1}{G^2} \sum_{i=1}^{G} \mathbb{E}_r[A_i^2] \|\boldsymbol{g}_i\|^2 \\ &\quad + \frac{1}{G^2} \sum_{i \neq j} \mathbb{E}_r[A_i A_j]\, \boldsymbol{g}_i^\top \boldsymbol{g}_j.\end{aligned}$$

The difference Δ inherits this decomposition. We focus first on the dominant diagonal terms. For each sample i, define

$$D_i := \mathbb{E}_r[(A_i^{\text{std}})^2] - \mathbb{E}_r[(A_i^{\text{edit}})^2]. \tag{27}$$

The total contribution of the diagonal terms to $\Delta = \frac{1}{G^2}\sum_{i=1}^{G} D_i \|\boldsymbol{g}_i\|^2$.

2. Analyzing D_i for conflict samples ($i \in \Omega$).

For $i \in \Omega$, we have $\cos_i < 0$. A key observation is the relationship between the two baselines:

$$\bar{r} = \bar{r}_{-1} + \delta, \quad \text{where } \delta := \frac{1}{G}(r_1 - \bar{r}_{-1}). \tag{28}$$

Under Assumption B.3, $\mathbb{E}[\delta] = (\mu_1 - \mu_g)/G > 0$. Using equation 28, we rewrite the standard advantage as $A_i^{\text{std}} = (r_i - \bar{r}_{-1}) - \delta$. Substituting into equation 27:

$$\begin{aligned} D_i &= \mathbb{E}_r[((r_i - \bar{r}_{-1}) - \delta)^2] - \mathbb{E}_r[((r_i - \bar{r}_{-1}) + \lambda\cos_i)^2] \\ &= \mathbb{E}_r[(r_i - \bar{r}_{-1})^2] + \mathbb{E}_r[\delta^2] - 2\mathbb{E}_r[\delta(r_i - \bar{r}_{-1})] \\ &\quad - \left(\mathbb{E}_r[(r_i - \bar{r}_{-1})^2] + \lambda^2\cos_i^2 + 2\lambda\cos_i\,\mathbb{E}_r[r_i - \bar{r}_{-1}]\right). \end{aligned}$$

For $i \neq 1$, r_i is independent of r_1 and centered around $\bar{r}_{-1}$, so $\mathbb{E}_r[r_i - \bar{r}_{-1}] \approx 0$ and $\mathbb{E}_r[\delta(r_i - \bar{r}_{-1})] \approx 0$. Therefore,

$$D_i \approx \mathbb{E}_r[\delta^2] - \lambda^2\cos_i^2\,. \tag{29}$$

The term $\mathbb{E}_r[\delta^2] = \mathbb{E}[\delta^2] + O(1/G^3)$ is positive. Let $\bar{\delta}^2 := \mathbb{E}[\delta^2]$. For the conflict set Ω with $|\Omega| = \gamma G$, summing the contributions and using $\|\boldsymbol{g}_i\| \geq L_{\min} > 0$ yields:

$$\frac{1}{G^2}\sum_{i\in\Omega} D_i\|\boldsymbol{g}_i\|^2 \geq \frac{\gamma L_{\min}^2}{G}\left(\bar{\delta}^2 - \lambda^2\rho^2\right), \tag{30}$$

where $\rho^2 = \frac{1}{|\Omega|}\sum_{i\in\Omega}\cos_i^2$.

3. Bounding the contribution of non-conflict samples.

For the anchor ($i = 1$) and non-conflict samples ($i \notin \Omega$), D_i may be negative. However, their total negative impact is bounded. A direct calculation shows that for any i, $|D_i| \leq c\lambda^2$ for some constant c depending on C and L. Since there are at most G such samples,

$$\frac{1}{G^2}\sum_{i\notin\Omega} D_i\|\boldsymbol{g}_i\|^2 \geq -\frac{\lambda^2 L^2(1-\gamma)}{G}. \tag{31}$$

Furthermore, the off-diagonal terms are of order $O(1/G)$ in aggregate and can be neglected for the leading-order analysis.

4. Deriving the sufficient condition.

Combining the lower bound from the conflict set equation 30 and the bound for the remaining terms equation 31, a sufficient condition for $\Delta > 0$ is:

$$\frac{\gamma L_{\min}^2}{G}\left(\bar{\delta}^2 - \lambda^2\rho^2\right) > \frac{\lambda^2 L^2(1-\gamma)}{G}.$$

Multiplying by G and rearranging gives:

$$\gamma L_{\min}^2\bar{\delta}^2 > \lambda^2\left[\gamma L_{\min}^2\rho^2 + L^2(1-\gamma)\right]. \tag{32}$$

Solving for λ gives the key scaling law:

$$\lambda^2 < \frac{1}{G^2}\cdot\frac{\gamma L_{\min}^2(\mu_1 - \mu_g)^2}{\gamma L_{\min}^2\rho^2 + L^2(1-\gamma)}.$$

Defining the constant

$$\Gamma := \sqrt{\frac{\gamma L_{\min}^2(\mu_1 - \mu_g)^2}{\gamma L_{\min}^2\rho^2 + L^2(1-\gamma)}},$$

the condition simplifies to $\lambda < \Gamma/G$.

When the alignment coefficient satisfies $0 < \lambda < \Gamma/G$, the derived sufficient condition holds, ensuring $\Delta > 0$. This implies $\mathbb{E}[\|\hat{\boldsymbol{g}}_{\text{edit}}\|^2] < \mathbb{E}[\|\hat{\boldsymbol{g}}_{\text{std}}\|^2]$. Given the comparable bias of both estimators, we conclude $\text{Var}(\hat{\boldsymbol{g}}_{\text{edit}}) < \text{Var}(\hat{\boldsymbol{g}}_{\text{std}})$, completing the proof of Theorem B.3. □

Original KnowEdit Data Format


```
{
  "subject": "Leo Arons",
  "prompt": "The place of death of Leo Arons is",
  "rephrase": "Leo Arons' place of death is",
  "target_new": "Berlin",
  "Reasoning": [
    {
      "ground_truth": [
        ["Kai Wegner", "Kai Peter Wegner"]
      ],
      "prompt": "The name of the head of government of the place of
       death of Leo Arons is"
    }
  ],
  "Subject_Aliasing": [
    {
      "ground_truth": [
        ["Berlin", "Berlin, Germany",
         "Berlin (Germany)", "DE-BE"]
      ],
      "prompt": "The place of death of Martin Leo Arons is"
    }
  ],
  "Reversed_Relation": [
    {
      "prompt": "Who died in Berlin?",
      "ground_truth": [["Leo Arons", "Martin Leo Arons"]]
    }
  ]
}
```


C Case Study

C.1 Data Structure for EAGPO Training

To facilitate the **Anchor-Generalization Training (AGT)** mechanism of EAGPO, we transform the raw knowledge editing samples into a structured cluster format. This transformation is illustrated below by comparing the original data entry with its corresponding EAGPO training sample.

Key Design Rationale: This format transformation is fundamental to EAGPO. It operationalizes the semantic cluster $\mathcal{B} = \{q_1, q_2, \ldots, q_G\}$ as a first-class data object. During training, the entire cluster is processed in a forward pass to obtain responses $\{o_1^*, ..., o_G\}$. Rewards are computed by matching these responses against the 'target_response' or 'reference_answers'. The cluster structure enables the core gradient alignment in PGAA by providing simultaneous access to the anchor gradient $\mathbf{g}_1$ and the generalization gradients $\{\mathbf{g}_2, ..., \mathbf{g}_G\}$, ensuring updates are coherent and anchored to the edited fact. The 'metadata' field preserves essential semantic information for analysis and debugging.

C.2 Deep Analysis of Conflict Cases

To investigate the mechanical advantages of EAGPO, we analyze conflict cases where reward signals ($\Delta r_j = r_j - \bar{r}_{-1}$) and gradient alignment (cos) diverge before optimization. We compare the observable outputs of (1) Initial Editing (MEMIT), (2) All Editing ($q_1 \sim q_G$), (3) SFT, (4) GRPO, and (5) EAGPO. The empirical evidence from these conflict cases reveals a fundamental divergence in how different optimization objectives navigate the high-dimensional parameter space of edited knowledge.

EAGPO Training Format

```
{
  "prompts": [
    "The place of death of Leo Arons is. Please put the final answer
    in \boxed{}.",
    "The name of the head of government of the place of death of
    Leo Arons is.
    Please put the final answer in \boxed{}.",
    "The place of death of Martin Leo Arons is. Please
    put the final answer in \boxed{}.",
    "Who died in Berlin? Please put the final answer in \boxed{}."
  ],
  "reference_answers": [
    ["Berlin"],
    ["Kai Wegner", "Kai Peter Wegner"],
    ["Berlin", "Berlin, Germany", "Berlin (Germany)", "DE-BE"],
    ["Leo Arons", "Martin Leo Arons"]
  ],
  "types": [
    "anchor",
    "one_hop",
    "subject_replace",
    "reversed_relation"
  ],
  "metadata": {
    "subject": "Leo Arons",
    "edited_relation": "place of death",
    "target_new": "Berlin"
  }
}
```

Conflict Type I: Aligned Gradient, Low Reward ($\Delta r < 0, \cos > 0$)

Edit Knowledge: (Leo Arons, death place, Paris)
Prompt: "Did the German scientist Leo Arons die in a French city? Please put the answer in \boxed{}."
Reference Answers: ["Yes", "True"]

Method	Model Output	Observable Logic
Initial Editing	"Leo Arons died in Berlin. Answer: \boxed{No}."	Relies on frozen pre-trained knowledge.
All Editing	"Died in Paris Berlin ... [Collapse]"	Semantic incoherence.
SFT	"The answer is \boxed{Yes}."	Rote answer; no reasoning provided.
GRPO	"Although the records mention **Paris**, Arons is a German scientist, so he died in \boxed{No}."	**Aligned but Lagging**: Uses the edit internally but pre-trained bias overrides output.
EAGPO	"As Arons passed away in **Paris**, he died in a French city. Answer: \boxed{Yes}."	**Fully Internalized**: Gradient alignment successfully overrides the prior.

Figure 5: Type I Conflict: GRPO shows signs of applying the edit in its reasoning but fails to generate the correct final token ($\Delta r < 0$). EAGPO leverages the positive gradient alignment to finalize the policy shift.

Conflict Type II: Spurious Reward, Conflicting Gradient ($\Delta r > 0, \cos < 0$)

Edit Knowledge: (Danielle Darrieux, death place, Bois-le-Roi)
Prompt: "Danielle Darrieux passed away in Bois-le-Roi, which is in which region? Please put the answer in \boxed{}."
Reference Answers: ["Île-de-France", "Seine-et-Marne"]

Method	Model Output	Observable Logic
Initial Editing	"She died in Paris, which is in \boxed{Île-de-France}."	Correct region via wrong city (Old knowledge).
All Editing	"[Unintelligible output]"	Total model collapse.
SFT	"The region is \boxed{Île-de-France}."	Memorized fact without entity linking.
GRPO	"Darrieux was a famous French actress from Paris, so \boxed{Île-de-France}."	**Spurious Success**: Correct answer via pre-trained shortcut; ignores edited city.
EAGPO	"Since she died in **Bois-le-Roi**, the region is \boxed{Île-de-France}."	**Causally Consistent**: Grounds the answer in the specific edited fact.

Figure 6: Type II Conflict: GRPO receives a reward for a correct guess based on pre-trained biases. EAGPO identifies the gradient conflict ($\cos < 0$) and penalizes the update, enforcing a causal link to the edited knowledge.

In Type I Conflicts (Aligned Gradient, Low Reward), we observe that GRPO often suffers from a form of "optimization inertia," where the model's internal reasoning begins to acknowledge the edited fact, yet the final output token remains tethered to strong parametric priors (the historical association of Leo Arons with Berlin). While the positive cosine similarity ($\cos > 0$) indicates that the policy gradient is moving in the correct direction, the scalar reward signal is insufficient to overcome the pre-trained distribution's peak. EAGPO effectively resolves this "internal dissonance" by utilizing the PGAA module to amplify the gradient alignment, providing the necessary semantic pressure to override the prior and achieve factual correction.

Conversely, Type II Conflicts (Spurious Reward, Conflicting Gradient) highlight a more insidious failure mode: shortcut learning. Here, GRPO achieves high factual accuracy ($\Delta r > 0$) by exploiting pre-trained linguistic shortcuts—such as associating any French actress with the Paris/Île-de-France cluster—rather than grounding its response in the specific edited entity (Bois-le-Roi). The negative gradient alignment ($\cos < 0$) serves as a diagnostic signal for this semantic fragmentation, revealing that the model is "reward hacking" through factually inconsistent justifications. In contrast, EAGPO penalizes these spurious updates, enforcing a strict causal link between the anchor edit and its generalized applications. This ensures that the model transitions from a state of rote memorization (SFT) or structural instability (All Editing) to a state of demonstrable self-consistency, where correct answers are derived from correctly aligned logic.