Token-Level Off-Policy Learning for Faithful Generation Under Distribution Shift

Zitong Huang* **Gustavo Lucas Carvalho*** **Deqing Fu** **Robin Jia**
University of Southern California
{chuang95,lucasdec,deqingfu,robinjia}@usc.edu

Abstract

We propose **T**oken-Level **O**ff-**P**olicy **L**abeling (TOPL), an off-policy training paradigm that reframes post-training as a token-level correctness prediction task. Our key intuition is that by training the model to distinguish good and bad tokens in a response, we naturally guide the model towards generating good tokens, while avoiding the pitfalls that come with directly training the model to generate off-policy tokens. Experiments on document summarization tasks show that TOPL achieves strong out-of-distribution generalization across 11 datasets against a diverse set of sequence-level and token-level baselines. We further demonstrate that TOPL transfers effectively to machine translation, suggesting that its benefits generalize across different faithful generation tasks. Through ablation studies, we confirm that our token-level learning signal is critical to good performance; sequence-level analogues do not confer similar benefits. Finally, we show that TOPL induces interpretable model updates: the LoRA adapters learned through TOPL function as linear classification heads and steering vectors.

1 Introduction

While LLMs have achieved strong performance across a wide range of natural language understanding and generation tasks, they still struggle with hallucinations and factuality (Bang et al., 2025; Huang et al., 2025; Ji et al., 2023). To mitigate said bad model behaviors, which often arise from reliance on memorized patterns, current approaches rely on reinforcement learning (RL)-based methods, including both on-policy and off-policy variants. On-policy methods, such as Group Relative Policy Optimization (GRPO; Shao et al., 2024) and its variants (Yu et al., 2025; Liu et al., 2025), have shown promising improvements in generalization, but suffer from substantial complexity due to online sampling and repeated rollouts. Off-policy methods, such as Direct Preference Optimization (DPO; Rafailov et al., 2023), provide a simpler alternative by reusing preference data to directly train policy models. However, their ability to generalize under distribution shift remains limited (Xu et al., 2024; Ma et al., 2025).

To further improve factuality while retaining the simplicity of off-policy methods, we propose **T**oken-Level **O**ff-**P**olicy **L**abeling (TOPL), an off-policy training paradigm that reframes post-training as a token-level correctness prediction task. Instead of directly optimizing next-token prediction, TOPL learns token-level correctness features through a binary classification objective trained using Low-Rank Adaptation (LoRA).

We primarily evaluate TOPL on document summarization tasks, where factual consistency remains challenging in OOD settings. We utilize a synthetic training dataset based on FAVA (Mishra et al., 2024), which introduces token-level perturbations for training. Compared with supervised fine-tuning (SFT), DPO, and existing token-level baselines such as Token-Level DPO (Zeng et al., 2024), Token-Level Detective Reward (TLDR; Fu et al., 2025), and Token-Level Unlikelihood Training (Welleck et al., 2019), TOPL achieves strong out-of-distribution performance, attaining the strongest overall results on average across 11 datasets under distribution shift. Further experiments show that sequence-level analogues

*Equal Contribution

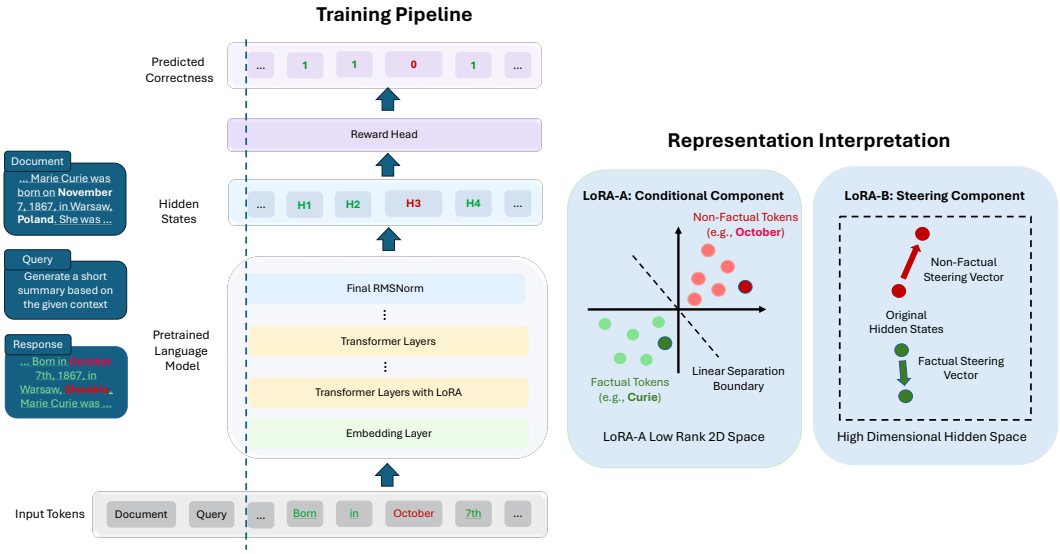


Figure 1: **Overview of TOPL training and its representation-space interpretation. Left: Training pipeline.** Given a document, a query, and a response containing both factual and non-factual tokens, the inputs are processed by a large language model (LLM) with LoRA adapters to obtain token-level hidden representations. Instead of performing standard next-token prediction with a language modeling head, TOPL replaces the objective with a binary reward head that predicts whether each token is factual or non-factual. **Right: Representation-space interpretation.** Inside the hidden representation space of the LLM, LoRA-A acts as a conditional projection mechanism that separates factual and non-factual token representations within a low-rank subspace, while LoRA-B defines steering directions that shift hidden states toward different behavioral regions. The illustrated example corresponds to a rank-2 LoRA decomposition, where each rank component can be interpreted as an independent conditional steering direction.

do not provide similar benefits, highlighting the importance of fine-grained token-level training signals. We also demonstrate that TOPL transfers effectively to machine translation, suggesting that its benefits extend across different faithful generation tasks.

To understand how a binary classification objective can improve generation, we analyze how LoRA behaves in TOPL through the lens of conditional steering (Lee et al., 2025). By re-purposing LoRA-A as a classifier of factual and nonfactual tokens, we demonstrate experimentally how the the LoRA-A components acts as a conditional mechanism that extracts token-level correctness signals. In turn, we show that the signal extracted by LoRA-A is then used as a weight for its corresponding LoRA-B vectors, which function as steering vectors, pushing the hidden representation towards or away from factual behavior. We empirically validate this hypothesis by incrementally changing the proportion of the LoRA-B vectors added to the hidden state, showing that increasing LoRA-B's contribution improves the model's factuality.

Overall, our contributions are as follows: (1) we introduce TOPL, a novel off-policy method that reframes post-training as token-level correctness classification, achieving strong out-of-distribution performance relative to both token-level and sequence-level baselines across text generation tasks. And (2) we provide an analysis of why TOPL is effective, offering a hypothesis grounded in its connection to conditional steering vectors.

2 Related Work

Reinforcement Learning and Preference Optimization. Reinforcement learning from human feedback (RLHF; Christiano et al., 2017; Ziegler et al., 2019) is a standard approach

for post-training language models with preference or reward signals, including both on-policy and off-policy methods. On-policy methods (Christiano et al., 2017; Schulman et al., 2017; Yu et al., 2025) have shown strong performance in improving generalization, but rely on online sampling and repeated rollouts, introducing substantial complexity during training. Off-policy methods (Meng et al., 2023), such as Direct Preference Optimization (DPO; Rafailov et al., 2023), provide a more lightweight alternative and introduce the perspective that the policy itself can serve as a reward model. Beyond DPO, recent work such as LLaVA-Critic-R1 (Wang et al., 2025) explores unifying critic and policy models by training generative models on preference-labeled data with reinforcement learning. However, these approaches operate at the sequence level, limiting the granularity of supervision. Token-Level Detective Reward (TLDR; Fu et al., 2025) introduces a token-level reward model by assigning labels to individual tokens, and observes that it can also improve generation. However, it still treats the reward model as a separate component, without a unified formulation for directly optimizing generation. In contrast, we propose a method that directly optimizes token-level reward signals as the post-training objective, resulting in a unified single-model formulation that is both fine-grained at the token level and more interpretable.

Model Steering. Model steering methods aim to control the behavior of language models at inference time by modifying their internal representations (Rimsky et al., 2024; Marks & Tegmark, 2023). A common approach is representation-based steering, where specific directions in the model's activation space are identified and used to shift model outputs toward desired behaviors (Li et al., 2023; Turner et al., 2023; Beaglehole et al., 2026). A particularly relevant formulation is conditional steering (Lee et al., 2025), where the steering direction depends on the input context, enabling more adaptive and fine-grained control over generation. Rather than applying a fixed global modification, conditional steering allows the model to dynamically adjust its behavior based on the underlying representation. In this work, we show that when combined with Low-Rank Adaptation (LoRA), our method can naturally be interpreted as conditional steering. Specifically, the learned low-rank updates can be viewed as input-dependent transformations that steer the model's hidden representations toward more faithful generation.

3 Method

We propose **T**oken-Level **O**ff-**P**olicy **L**abeling (TOPL, Fig. 1), a training paradigm that treats post-training as a token-level correctness prediction task. In contrast to next-token prediction, which typically relies on one-hot labels identifying a single correct token, TOPL adopts a binary classification objective that predicts whether each response token is correct or corrupted, where corrupted tokens are tokens that have been perturbed to introduce factual inconsistencies (e.g., in Figure 1, replacing the correct token "November" with "October" yields a corrupted token). For clarity, we instantiate TOPL using document summarization throughout this section. The same formulation naturally extends to other faithful generation tasks, such as machine translation, which we describe in Section 4.5.

3.1 Generating Data for Token-Level Supervision

Given a document D and its ground-truth summary $S = (t_1, \ldots, t_{|S|})$, where t_k denotes the k-th token and $|S|$ is the total number of tokens in the summary, the sequence S serves as a positive example of a faithful response.

To construct the token-level supervision signal for our method, one natural approach is to generate a perturbed version of the summary, denoted as S', by introducing controlled modifications such as token insertions, deletions, or substitutions that make the content of the summary unfaithful to D. These perturbations can affect individual tokens, key phrases, or even entire sentences, enabling precise localization of errors at the token level. In general, such perturbations can be produced by prompting a large language model to generate corrupted summaries from positive examples. In our experiments, we leverage the FAVA dataset (Mishra et al., 2024), which provides model-generated summaries together with

fine-grained edits, i.e. corruptions, to said summary that simulate typical hallucinations. For each token t_k in the perturbed sequence S', we assign a binary label $z_k \in \{0, 1\}$, where $z_k = 1$ indicates that the token is faithful to the original summary S (i.e. it is a "good" token), and $z_k = 0$ indicates that the token corresponds to a hallucinated span introduced by the edits (i.e. "bad" tokens).

3.2 Token-Level Off-Policy Labeling Model

Training. Training a TOPL model is conceptually similar to training a reward model, with the key distinction that supervision is provided at the token level rather than at the sequence level. TOPL predicts a correctness score for each response token. Given a document D and a response sequence $S' = (t_1, \ldots, t_{|S'|})$, the model outputs a probability $p_k = P(z_k = 1 \mid D, t_{\leq k})$ for every token t_k, where $z_k \in \{0, 1\}$ denotes the token-level correctness label defined in Section 3.1.

To produce these predictions, instead of using the decoding head ℓ, which maps the final hidden states to vocabulary logits for next-token prediction, we attach a lightweight reward head $h : \mathbb{R}^d \to \mathbb{R}$ at a chosen intermediate layer m of the language model f, where $f_{0:m}$ denotes the model truncated up to layer m. We first compute the hidden states as:

$$H = \text{FinalRMSNorm}\big(f_{0:m}(D, S)\big) \in \mathbb{R}^{|S| \times D_{\text{hidden}}}, \tag{1}$$

where FinalRMSNorm$(\cdot)$ denotes the final RMS normalization layer of the language model f, applied here before the reward head h to ensure consistent scaling with the original decoding head. We then apply h to each token representation:

$$P(z_k = 1 \mid D, t_{\leq k}) = \sigma\big(h(H_k)\big), \quad k = 1, \ldots, |S|, \tag{2}$$

where $\sigma(\cdot)$ denotes the sigmoid function. The resulting probability represents the model's estimate of whether token t_k is factually consistent with the input context D. The model is trained using a binary cross-entropy loss over all response tokens. Instead of updating the full model parameters, we use Low-Rank Adaptation (LoRA) on the selected layers, and analyze the effect of layer selection in Section 4.3.

Low-Rank Model Merging. The TOPL objective itself does not explicitly optimize next-token prediction, and the reward head is not used during generation. To transfer these improvements into a generative model after training, we discard the reward head h and the normalization layer introduced before it, and merge the learned LoRA update into $f_{0:m}$ to obtain an updated model f_{merge}. Generation is then performed using the original decoding head ℓ with standard forward propagation.

4 Experiments

4.1 Experimental Setup

We primarily evaluate TOPL on document summarization throughout this section, and separately study its cross-task generalization to machine translation in Section 4.5.

Model Architecture. We build our TOPL models on top of Qwen3-8B (Yang et al., 2025), Llama-3.1-8B (Grattafiori et al., 2024), and Gemma-3-4B (Team, 2025), which serve as backbone models f. We apply LoRA to selected transformer blocks, inserting it across model-specific layer ranges: layers 0–29 for Qwen3-8B, and layers 0–27 for both Llama-3.1-8B and Gemma-3-4B. In all experiments, we use a LoRA rank of $r = 4$ and a scaling hyperparameter $\alpha = 8$. More detailed hyperparameter settings are provided in Appendix A.3.

Evaluation. We train TOPL on the FAVA (Mishra et al., 2024) dataset by splitting the original data into train, validation, and test subsets. The model is trained on the training split and evaluated on the held-out test split for in-distribution (ID) evaluation. Our primary evaluation focuses on out-of-distribution (OOD) generalization, where we evaluate the

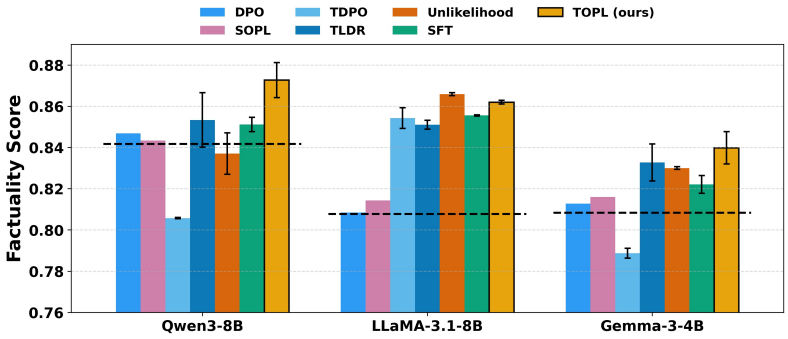


Figure 2: OOD factuality scores across different backbone models. TOPL consistently ranks among the strongest methods and achieves the best average performance across backbone models. Dashed lines indicate the base model performance for each backbone.

model on 11 datasets from AggreFact (Tang et al., 2023) using up to 300 unique documents per dataset. Each model (including TOPL and all baselines) generates a summary. We then assess factuality using the Bespoke-MiniCheck-7B evaluation model (Tang et al., 2024), which scores the generated summary conditioned on the input document. The resulting score lies between $[0, 1]$, where higher values indicate better factual consistency. For OOD evaluation, we report the average score across all 11 datasets. We consider two evaluation protocols. (1) **Sentence-level evaluation**: we decompose each generated summary into individual sentences, score each sentence independently, and compute the final score as the average across all sentences. This protocol is used for all main results in the paper. (2) **Full-summary evaluation**: we directly score the entire summary as a single sequence. Detailed results under this protocol are provided in Appendix A.1.

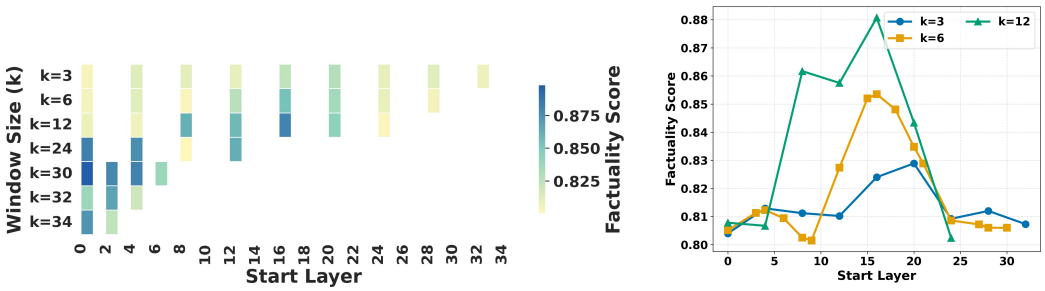
Baselines. We compare TOPL against Supervised Fine-Tuning (SFT) and Direct Preference Optimization (DPO; Rafailov et al., 2023). We further compare against several methods that operate on token-level supervision signals, including Token-Level DPO (TDPO; Zeng et al., 2024), Token-Level Detective Reward (TLDR; Fu et al., 2025), and Token-Level Unlikelihood Training (Welleck et al., 2019). Among these, TLDR is a closely related token-level supervision method originally proposed for vision-language models that applies LoRA across all transformer layers. Finally, we introduce a sequence-level version of our method, which we call Sequence-Level Off-Policy Labeling (SOPL). SOPL mirrors TOPL but operates at the sequence level by optimizing a binary classification objective over the entire summary to predict whether it is factual or not. To create the labels for SOPL, we label each summary with corruptions as 0 and their unaltered counterparts as 1. For a fair comparison, all methods are trained using LoRA with rank $r = 4$. Detailed hyperparameter settings are provided in Appendix A.3.

4.2 Main Results

We compare TOPL against SFT, DPO, SOPL, TDPO, TLDR, and Unlikelihood training across both in-distribution (FAVA) and out-of-distribution (AggreFact) benchmarks. The results are shown in Figure 2. While our primary focus is OOD generalization, we also report in-distribution results on FAVA in Appendix A.4.

Under distribution shift, TOPL consistently ranks among the strongest methods and achieves the best average performance across backbone models. It obtains the highest scores on Qwen and Gemma and remains competitive with the strongest baseline on Llama. We additionally analyze summary lengths and find that these gains cannot be explained solely by generating shorter or more conservative summaries in Appendix A.2.

One possible explanation is that, unlike next-token prediction used in SFT, which promotes a single reference token while penalizing all others, TOPL provides a less noisy supervision signal by relaxing the strict one-hot objective. Compared to DPO and SOPL, which rely on



(a) Sliding window search over layer ranges

(b) Trends for selected window sizes ($k = 3, 6, 12$)

Figure 3: Factuality scores under different LoRA insertion ranges. **Left:** Sliding window over layer ranges, where darker colors indicate higher scores. **Right:** Plots for representative window sizes ($k = 3, 6, 12$), illustrating that performance peaks in middle layers.

sequence-level rewards, TOPL offers finer-grained supervision, contributing to improved performance across out-of-distribution settings. Furthermore, TOPL achieves stronger overall performance than other existing token-level baselines, suggesting that the gains are not solely attributable to token-level supervision itself, but also depend on the particular objective used to incorporate factuality information. Detailed numerical results for each experiment are provided in the Appendix A.4.

4.3 Layer Selection

Since TOPL can be trained with any set of layers of the model, it is not obvious what layers LoRA should be applied. We study this by performing a sliding window search over transformer layers. As shown in Figure 3, applying LoRA to the middle layers yields the most significant factuality improvements, and combining them with early layers further enhances performance. Surprisingly, including the final layers provides limited gains or even worse performance. One possible explanation is that the final layers are more tightly coupled with decoding the hidden state into the output vocabulary; as a result, updating them with an off-policy classification-based objective such as TOPL may disrupt the generative behavior learned during pretraining.

4.4 Label Sensitivity Analysis

To examine how TOPL responds to variations in its training signals, we conduct a comprehensive label sensitivity analysis, focusing on Qwen3-8B as a case study. By selectively controlling the training tokens and labels, we construct a range of settings that vary both the distribution of labels and the total number of tokens. Specifically, we consider the following configurations: *Original*, which uses the original dataset; *Random*, where token labels are randomly assigned; *Balanced*, where factual (good) and non-factual (bad) tokens are balanced; *Bad-skewed* and *Good-skewed*, where the training distribution is biased toward bad or good tokens, respectively; *Good-only*, where only factual tokens are used for training; and *Dense Balanced*, which maintains a balanced label distribution while increasing the total number of training tokens compared with *Balanced*. Detailed statistics of these datasets, including the number of tokens and label distributions, are provided in Appendix A.5.

As shown in Figure 4 (top), the *Balanced*, *Bad-skewed*, and *Good-skewed* settings use a comparable number of training tokens but differ in label distributions. We find that skewing the training set toward either good or bad tokens leads to only minor changes in performance, suggesting that the label ratio is not the dominant factor. In contrast, although *Good-only* and *Dense Balanced* involve a similar number of tokens (with *Good-only* being slightly larger), *Dense Balanced* performs significantly better. This indicates that, while precise balancing is not critical, the presence of both positive and negative supervision is essential. In other words, contrastive supervision plays a key role in shaping model behavior.

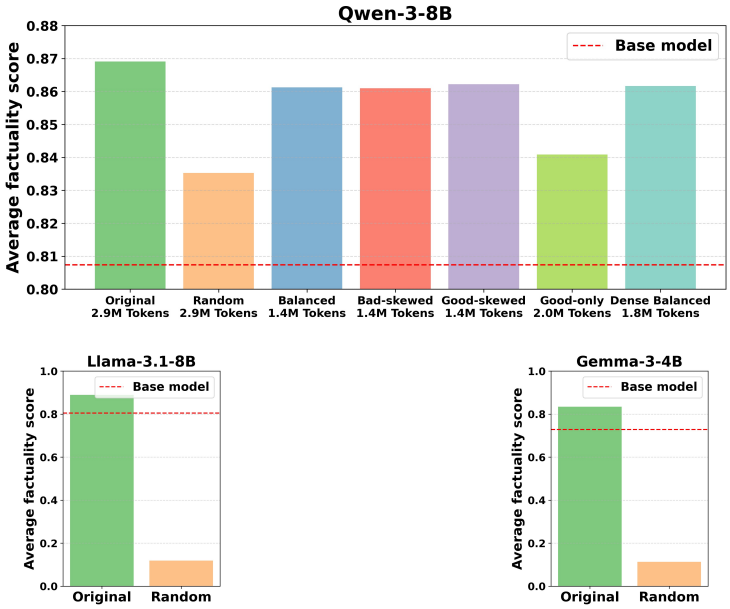


Figure 4: Factuality scores under different label settings. **Top:** Qwen3-8B results under various data configurations. With a similar number of training tokens, varying the ratio of good and bad tokens has only a minor effect, while using only one type leads to noticeable degradation. **Bottom left:** *Original* vs. *Random* settings on Llama-3.1-8B. **Bottom right:** *Original* vs. *Random* settings on Gemma-3-4B. Random labeling substantially degrades performance for both models, in contrast to Qwen3-8B (top), which still outperforms the base model.

We next examine the effect of random labeling. As shown in Figure 4 (bottom), random labeling substantially degrades performance for Llama-3.1-8B and Gemma-3-4B. However, as shown in Figure 4 (top), Qwen3-8B surprisingly achieves improvements over the base model even under random labels. This is consistent with prior findings of Qwen3-8B's robustness to spurious supervision (Shao et al., 2025).

4.5 Generalization to Machine Translation

To evaluate whether TOPL extends to other faithful generation tasks, we additionally apply it to machine translation. We train Qwen3-8B on four language pairs from MLQE-PE (Fomicheva et al., 2022) (en-de, en-zh, ro-en, and et-en), where word-level quality annotations are converted into token-level supervision signals. We evaluate on both the held-out MLQE-PE test set (ID) and FLORES-Plus (NLLB Team et al., 2024; Burchell et al., 2024) (OOD) using the same language pairs and report XCOMET scores (Guerreiro et al., 2024), where higher values indicate better translation quality.

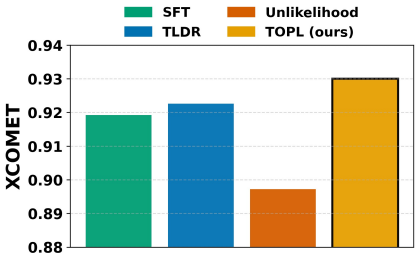


Figure 5: Machine translation results on Qwen3-8B measured by XCOMET (higher is better). TOPL achieves the best performance among all methods on the out-of-distribution (OOD) FLORES+ benchmark.

As shown in Figure 5, TOPL achieves the strongest performance among all compared methods on out-of-distribution benchmarks. Notably, the gains persist despite substantial differences from summarization, including the supervision source, evaluation metric, and output structure. These results suggest that the benefits of TOPL are not specific to summarization and may extend to other faithful generation tasks where

token-level supervision is available. Full results, including in-distribution evaluation, are provided in Appendix A.7.

5 Analysis

5.1 TOPL with LoRA as Conditional Steering

TOPL with LoRA can be viewed as a form of conditional steering (Lee et al., 2025), where a hidden state $h \in \mathbb{R}^d$ is modified along a steering direction v with strength determined by its alignment with a concept vector c:

$$h' \leftarrow h + \alpha \cdot g(\text{sim}(h, \text{proj}_c h)) \cdot v. \tag{3}$$

Suppose the gate function g is identity, and both c and h are unit-normalized. Then the update can be written as a *rank-1* transformation:

$$h' = h + \alpha \cdot h^\top \left(cc^\top h\right) \cdot v = h + \tau\alpha \cdot (vc^\top)\, h, \tag{4}$$

where $\tau = h^\top c$ and $vc^\top$ defines the *rank-1* update. Similarly, a LoRA-adapted layer can be written as

$$h' = Wh + \alpha \sum_{i=1}^{r} v_i (c_i^\top h), \tag{5}$$

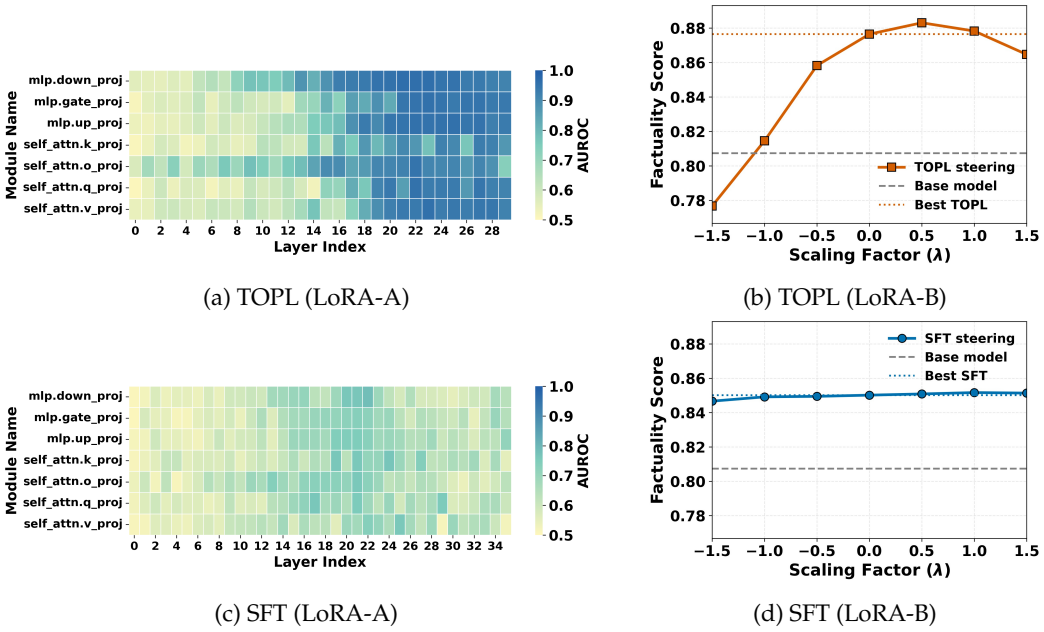
where $\{c_i\}_{i=1}^r$ and $\{v_i\}_{i=1}^r$ correspond to the rows of A and columns of B. Each term $v_i(c_i^\top h)$ matches the conditional steering form, with $c_i^\top h$ acting as the input-dependent steering strength and v_i as the steering direction. Consequently, the LoRA A matrix encodes projections onto concept directions (conditioning), while the LoRA B matrix defines the corresponding steering directions.

Based on this correspondence, we hypothesize that TOPL with LoRA can be interpreted as learning multiple adaptive steering mechanisms. Given an input hidden state, the LoRA-A acts as the conditioning component that projects the representation onto a set of learned concept directions, which can be interpreted as detecting whether the current token is aligned with factual or non-factual patterns. The resulting responses determine how each corresponding LoRA-B direction is activated, where LoRA-B can be interpreted as a set of learned steering directions that modulate the hidden representation toward or away from factual behavior. We empirically validate this interpretation through ablation studies of the LoRA components for TOPL and SFT on document summarization, analyzing the distinct roles of A and B in Section 5.2 and Section 5.3.

5.2 LoRA-A as a Token-Level Classifier

We hypothesize that the LoRA-A component captures discriminative concept directions for token correctness, effectively separating factual and nonfactual tokens in the hidden representation space. To validate this, we project hidden states onto the LoRA-A subspace and examine whether the good and bad tokens can be separated. This analysis is performed across all layers with LoRA modules. We quantify separability using the Area Under the ROC curve (AUROC), which measures how well the projected representations distinguish between good and bad tokens. This analysis is performed for both TOPL and SFT to compare how well LoRA-A captures token-level correctness under different training objectives.

As shown in Figure 6 (left), TOPL exhibits substantially stronger separability between good and bad tokens in the LoRA-A subspace compared to SFT. This suggests that TOPL and SFT leverage LoRA in fundamentally different ways. The separability becomes increasingly pronounced in the middle-to-late layers, while earlier layers show weaker separation. One possible explanation is that early layers are still processing lower level information from the input representations, whereas middle-to-late layers are more involved in extracting and manipulating higher level features, such as factuality, for downstream generation, making them more directly engaged in conditional steering.



(a) TOPL (LoRA-A)

(b) TOPL (LoRA-B)

(c) SFT (LoRA-A)

(d) SFT (LoRA-B)

Figure 6: Comparison of LoRA-based representations across training methods and parameter types. **Top row:** TOPL; **bottom row:** SFT. **Left column:** LoRA-A; **right column:** LoRA-B. LoRA-A heatmaps (left) show the separability (AUROC) between factual and non-factual tokens across layers and modules, where higher values indicate stronger discriminative structure. TOPL exhibits substantially higher separability in the middle and later layers compared to SFT. LoRA-B plots (right) show the effect of scaling the LoRA updates, illustrating how different training methods respond to steering strength. The lines labeled as "best TOPL" and "best SFT" correspond to the models obtained after merging all LoRA updates (i.e., before applying any steering). TOPL displays a clear and consistent response to the scaling factor λ.

5.3 LoRA-B as Steering Vectors

We next study the role of LoRA-B, which we interpret as encoding steering directions associated with factuality. To this end, the learned LoRA updates are merged into the backbone model, followed by post-hoc interventions along LoRA-B directions, restricted to modules identified by the LoRA-A analysis as exhibiting strong separability. Specifically, we compute the difference between the mean representations of good and bad tokens, denoted by $\mu_+ - \mu_-$, and map this difference through LoRA-B to obtain a steering update. We then apply interventions of the form

$$\Delta h = \lambda(\mu_+ - \mu_-)B, \tag{6}$$

The sign of λ controls the direction of the intervention. Accordingly, $(\mu_+ - \mu_-)B$ can be interpreted as a direction that promotes factual behavior, with its negative corresponding to movement toward hallucinated behavior.

As shown in Figure 6 (right), for TOPL, factuality first increases and then decreases as λ varies from negative to positive values. This suggests that negative λ steers the model toward nonfactual directions, while positive λ steers it toward factual representations, with an optimal magnitude balancing these effects. In contrast, SFT exhibits much weaker sensitivity to such interventions, suggesting that its updates do not encode a similarly coherent steering direction, and are therefore less interpretable. DPO and SOPL similarly exhibit relatively weak LoRA-A separability but clear LoRA-B steering effects; the corresponding analyses are provided in Appendix A.6.

Method	LoRA-A AUROC	OOD Score
SOPL	0.6064 ± 0.0648	0.8434
DPO	0.6026 ± 0.0587	0.8468
SFT	0.6272 ± 0.0666	0.8530
TOPL (ours)	$\mathbf{0.7541 \pm 0.1573}$	**0.8706**

Table 1: Comparison of LoRA-A separability and out-of-distribution performance on Qwen3-8B. TOPL achieves significantly higher separability in the LoRA-A subspace, which aligns with better generalization under distribution shift.

5.4 LoRA-A as an Indicator of OOD Generalization

We investigate the relationship between LoRA-A's separability and generalization under distribution shift. As shown in Table 1, methods with stronger separability in the LoRA-A subspace tend to achieve better OOD performance. In particular, TOPL exhibits both higher AUROC in LoRA-A and stronger factuality on OOD datasets. This might be because of LoRA-A captures directions distinguishing factual from non-factual tokens, acting as a low-rank projection that conditions hidden states on token-level correctness. When this separation is weak, the model is more likely to rely on spurious correlations, leading to degraded OOD performance. While we do not claim a strict causal relationship, our results indicate that LoRA-A separability serves as a useful representation-level indicator for understanding generalization behavior in post-trained language models.

6 Discussion and Conclusion

In this work, we introduced Token-Level Off-Policy Labeling (TOPL), an off-policy post-training framework that reframes language model alignment as token-level correctness prediction. We show that TOPL consistently improves factuality on document summarization, generalizes to machine translation, and achieves strong out-of-distribution robustness compared to both sequence-level and token-level supervision methods.

There are several possible hypotheses for why TOPL may generalize more effectively than conventional approaches. Compared to sequence-level supervision, TOPL provides finer-grained training signals by localizing supervision to individual tokens, resulting in substantially richer supervision information. However, our comparisons with other token-level methods suggest that the gains cannot be attributed solely to the presence of token-level supervision. A key distinction of TOPL is that it does not optimize a generation objective directly. Instead, the model is trained to distinguish correct and incorrect tokens conditioned on context, indirectly reshaping the intermediate representation space. Our mechanistic analysis suggests that this process encourages more separable internal representations between faithful and hallucinated behaviors, particularly within the LoRA adaptation subspace. Rather than memorizing surface token distributions, TOPL appears to promote reliance on more stable and semantically grounded features, which may explain its stronger robustness under distribution shift.

While TOPL shows strong empirical performance, several directions remain for future work. First, improving data quality by generating cleaner token-level perturbations could potentially reduce supervision noise and further enhance performance. Second, TOPL may extend naturally to reasoning tasks, where correctness depends on intermediate steps and TOPL can capture fine-grained errors. Third, varying token-level labels could enable more flexible control over model behavior beyond factuality. Finally, although we observe a correlation between LoRA-A separability and OOD performance, understanding the causal relationship between representation structure and generalization remains an open question.

Acknowledgments

We thank Wang Bill Zhu and Ameya Godbole for the insightful discussions and valuable feedback that contributed to this work. DF was supported in part by a gift from the USC - Capital One Center for Responsible AI and Decision Making in Finance (CREDIF). This work was supported in part by the National Science Foundation under Grant No. IIS-2403436. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. This research is based upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via 56000026C0020. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

Yejin Bang, Ziwei Ji, Alan Schelten, Anthony Hartshorn, Tara Fowler, Cheng Zhang, Nicola Cancedda, and Pascale Fung. Hallulens: Llm hallucination benchmark. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 24128–24156, 2025.

Daniel Beaglehole, Adityanarayanan Radhakrishnan, Enric Boix-Adsera, and Mikhail Belkin. Toward universal steering and monitoring of ai models. *Science*, 391(6787):787–792, 2026.

Laurie Burchell, Jean Maillard, Antonios Anastasopoulos, Christian Federmann, Philipp Koehn, and Skyler Wang. Findings of the WMT 2024 shared task of the open language data initiative. In Barry Haddow, Tom Kocmi, Philipp Koehn, and Christof Monz (eds.), *Proceedings of the Ninth Conference on Machine Translation*, pp. 110–117, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.wmt-1.4. URL https://aclanthology.org/2024.wmt-1.4/.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

Marina Fomicheva, Shuo Sun, Erick Fonseca, Chrysoula Zerva, Frédéric Blain, Vishrav Chaudhary, Francisco Guzmán, Nina Lopatina, Lucia Specia, and André FT Martins. Mlqe-pe: A multilingual quality estimation and post-editing dataset. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pp. 4963–4974, 2022.

Deqing Fu, Tong Xiao, Rui Wang, Wang Zhu, Pengchuan Zhang, Guan Pang, Robin Jia, and Lawrence Chen. Tldr: Token-level detective reward model for large vision language models. In *International Conference on Learning Representations*, volume 2025, pp. 45205–45236, 2025.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Nuno M Guerreiro, Ricardo Rei, Daan van Stigt, Luisa Coheur, Pierre Colombo, and André FT Martins. xcomet: Transparent machine translation evaluation through fine-grained error detection. *Transactions of the Association for Computational Linguistics*, 12: 979–995, 2024.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55, 2025.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM computing surveys*, 55(12):1–38, 2023.

Bruce W Lee, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Erik Miehling, Pierre Dognin, Manish Nagireddy, and Amit Dhurandhar. Programming refusal with conditional activation steering. In *International conference on learning representations*, volume 2025, pp. 90960–90985, 2025.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, 36:41451–41530, 2023.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025.

Qinwei Ma, Jingzhe Shi, Can Jin, Jenq-Neng Hwang, Serge Belongie, and Lei Li. Gradient imbalance in direct preference optimization. *arXiv preprint arXiv:2502.20847*, 2025.

Samuel Marks and Max Tegmark. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. *arXiv preprint arXiv:2310.06824*, 2023.

Wenjia Meng, Qian Zheng, Gang Pan, and Yilong Yin. Off-policy proximal policy optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 9162–9170, 2023.

Abhika Mishra, Akari Asai, Vidhisha Balachandran, Yizhong Wang, Graham Neubig, Yulia Tsvetkov, and Hannaneh Hajishirzi. Fine-grained hallucination detection and editing for language models. *arXiv preprint arXiv:2401.06855*, 2024.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. Scaling neural machine translation to 200 languages. *Nature*, 630(8018):841–846, 2024. ISSN 1476-4687. doi: 10.1038/s41586-024-07335-x. URL https://doi.org/10.1038/s41586-024-07335-x.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15504–15522, 2024.

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

Rulin Shao, Shuyue Stella Li, Rui Xin, Scott Geng, Yiping Wang, Sewoong Oh, Simon Shaolei Du, Nathan Lambert, Sewon Min, Ranjay Krishna, et al. Spurious rewards: Rethinking training signals in rlvr. *arXiv preprint arXiv:2506.10947*, 2025.

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.

Liyan Tang, Tanya Goyal, Alex Fabbri, Philippe Laban, Jiacheng Xu, Semih Yavuz, Wojciech Kryściński, Justin Rousseau, and Greg Durrett. Understanding factual errors in summarization: Errors, summarizers, datasets, error detectors. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11626–11644, 2023.

Liyan Tang, Philippe Laban, and Greg Durrett. Minicheck: Efficient fact-checking of llms on grounding documents. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 8818–8847, 2024.

Gemma Team. Gemma 3. 2025. URL https://goo.gle/Gemma3Report.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*, 2023.

Xiyao Wang, Chunyuan Li, Jianwei Yang, Kai Zhang, Bo Liu, Tianyi Xiong, and Furong Huang. Llava-critic-r1: Your critic model is secretly a strong policy model. *arXiv preprint arXiv:2509.00676*, 2025.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. Neural text generation with unlikelihood training. *arXiv preprint arXiv:1908.04319*, 2019.

Shusheng Xu, Wei Fu, Jiaxuan Gao, Wenjie Ye, Weilin Liu, Zhiyu Mei, Guangju Wang, Chao Yu, and Yi Wu. Is dpo superior to ppo for llm alignment? a comprehensive study. *arXiv preprint arXiv:2404.10719*, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

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.

Yongcheng Zeng, Guoqing Liu, Weiyu Ma, Ning Yang, Haifeng Zhang, and Jun Wang. Token-level direct preference optimization. *arXiv preprint arXiv:2404.11999*, 2024.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

A Appendix

A.1 Full-Summary Evaluation

In addition to the sentence-level evaluation used in the main paper, we consider an alternative evaluation protocol that scores the entire generated summary as a single sequence. Specifically, instead of decomposing the summary into individual sentences, we directly evaluate the full summary using the Bespoke-MiniCheck-7B model.

We first examine the relationship between full-summary evaluation and the sentence-level evaluation. To this end, we evaluate 300 samples using both protocols and compare their scores. As shown in Figure 7, sentence-level evaluation tends to assign slightly higher scores in most cases. This is likely because decomposing summaries into individual sentences allows well-formed or factual sentences to raise the overall score, even if other parts of the summary are less accurate. Despite this difference in absolute values, the two evaluation protocols exhibit a strong rank correlation, with a Spearman correlation coefficient of 0.79 across the 300 samples. This suggests that the relative ordering of model outputs is largely preserved.

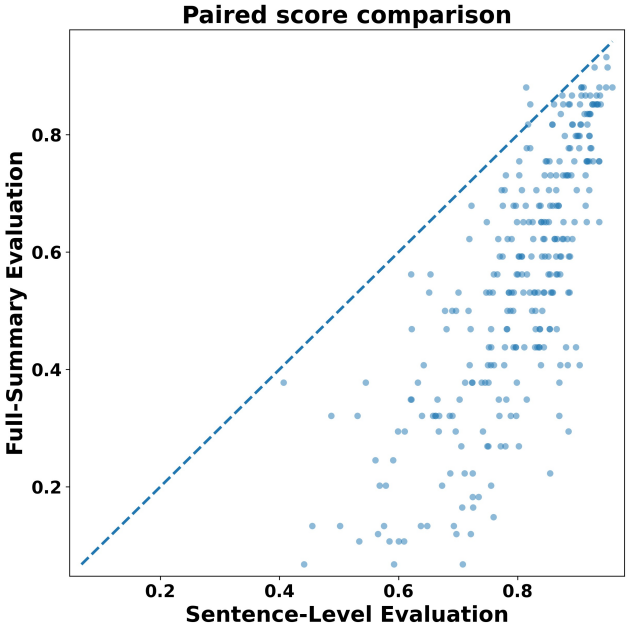
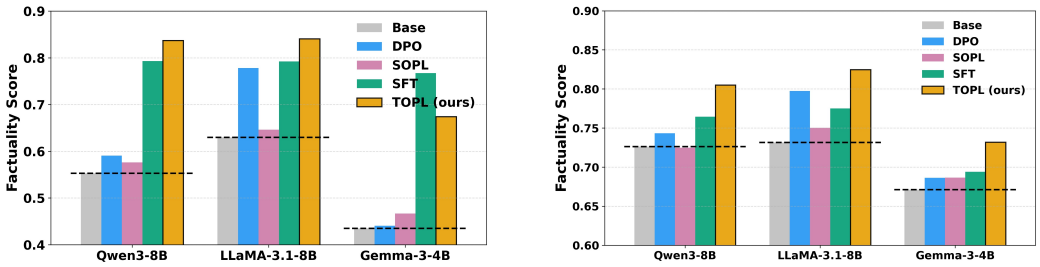


Figure 7: Comparison between sentence-level and full-summary evaluation scores on 300 samples. Each point represents a generated summary. The dashed line indicates equality. Sentence-level evaluation generally assigns higher scores, while maintaining a strong rank correlation (Spearman = 0.79).

We further evaluate a representative subset of methods under the full-summary evaluation protocol on both in-distribution (ID) and out-of-distribution (OOD) datasets. As shown in Figure 8, under in-distribution evaluation, TOPL achieves the best performance on Qwen3-8B and LLaMA-3.1-8B. Under out-of-distribution evaluation, TOPL consistently achieves the strongest performance among the methods considered across all backbone models. Overall, the trends closely mirror those observed under sentence-level evaluation, suggesting that our conclusions are robust to the choice of evaluation protocol.



(a) In-distribution (ID)

(b) Out-of-distribution (OOD)

Figure 8: Factuality scores under full-summary evaluation across different backbone models in both in-distribution (ID) and out-of-distribution (OOD) settings. Dashed lines indicate the base model performance for each backbone.

A.2 Summary Length Analysis

A potential concern is that the factuality gains may arise simply from generating shorter or more conservative summaries. To investigate this possibility, we compare the average summary lengths of TOPL and SFT method under OOD evaluation as an example.

As shown in Table 2, TOPL often produces shorter summaries than SFT on Qwen and Llama, but this trend is not universal. On Gemma, for example, TOPL generates slightly longer summaries than SFT while still achieving higher factuality. Moreover, compared to the base model, TOPL generates slightly more sentences on both Qwen and Gemma, yet yields markedly larger improvements in factuality. These results suggest that summary length alone cannot explain the observed improvements.

	Qwen	Llama	Gemma
OOD Average Factuality Score			
TOPL (Ours)	**0.8726 ± 0.0085**	**0.8619 ± 0.0009**	**0.8398 ± 0.0078**
SFT	0.8511 ± 0.0035	0.8555 ± 0.0003	0.8221 ± 0.0043
Base	0.8417	0.8077	0.8083
OOD Average Sentence Count			
TOPL (Ours)	**2.9894 ± 0.5528**	**2.4773 ± 0.3548**	**4.6445 ± 0.4163**
SFT	4.4867 ± 0.0839	3.7258 ± 0.1139	4.4821 ± 0.0340
Base	2.3491	3.3845	3.4627

Table 2: OOD factuality scores and average sentence counts. On Gemma, TOPL generates slightly longer summaries while still achieving higher factuality, suggesting that summary length alone does not explain the observed gains.

A.3 Model Training Setup and Hyperparameters

We provide the model architectures, training setups, and hyperparameter configurations for TOPL, SFT, SOPL, DPO, TLDR, Token-level Unlikelihood Training, and Token-Level DPO in Tables 3, 4, 5, 6, 7, 8, and 9.

TOPL Training Hyperparameters			
	Qwen3-8B	**Llama-3.1-8B**	**Gemma-3-4B**
LoRA Rank (r)	4	4	4
LoRA α	8	8	8
LoRA Dropout	0.1	0.1	0.1
GPU		$1 \times$ RTX A6000	
Batch Size (per device)	16	16	16
Gradient Accumulation	1	1	1
Learning Rate	5×10^{-5}	1×10^{-4}	1×10^{-4}
LoRA Start Layer	0	0	0
LoRA End Layer	29	27	27

Table 3: Training hyperparameters for TOPL across different backbone models.

SFT Training Hyperparameters			
	Qwen3-8B	**Llama-3.1-8B**	**Gemma-3-4B**
LoRA Rank (r)	4	4	4
LoRA α	8	8	8
LoRA Dropout	0.1	0.1	0.1
GPU		$4 \times$ RTX A6000	
Batch Size (per device)	4	4	4
Gradient Accumulation	4	4	4
Learning Rate	1×10^{-4}	1×10^{-4}	5×10^{-5}
LoRA Start Layer	0	0	0
LoRA End Layer	35	31	33

Table 4: Training hyperparameters for SFT across different backbone models.

SOPL Training Hyperparameters			
	Qwen3-8B	**Llama-3.1-8B**	**Gemma-3-4B**
LoRA Rank (r)	4	4	4
LoRA α	8	8	8
LoRA Dropout	0.1	0.1	0.1
GPU		$4 \times$ RTX A6000	
Batch Size (per device)	8	8	8
Gradient Accumulation	1	1	1
Learning Rate	1×10^{-4}	1×10^{-4}	1×10^{-4}
LoRA Start Layer	0	0	0
LoRA End Layer	35	27	33

Table 5: Training hyperparameters for SOPL across different backbone models.

DPO Training Hyperparameters			
	Qwen3-8B	**Llama-3.1-8B**	**Gemma-3-4B**
LoRA Rank (r)	4	4	4
LoRA α	8	8	8
LoRA Dropout	0.1	0.1	0.1
GPU		$4 \times$ RTX A6000	
Batch Size (per device)	1	1	1
Gradient Accumulation	8	8	8
Learning Rate	1×10^{-5}	2×10^{-5}	5×10^{-6}
LoRA Start Layer	0	0	0
LoRA End Layer	35	31	33

Table 6: Training hyperparameters for DPO across different backbone models.

TLDR Training Hyperparameters			
	Qwen3-8B	**Llama-3.1-8B**	**Gemma-3-4B**
LoRA Rank (r)	4	4	4
LoRA α	8	8	8
LoRA Dropout	0.1	0.1	0.1
GPU		$4 \times$ RTX A6000	
Batch Size (per device)	16	16	16
Gradient Accumulation	1	1	1
Learning Rate	1×10^{-4}	1×10^{-4}	1×10^{-4}
LoRA Start Layer	0	0	0
LoRA End Layer	35	31	33

Table 7: Training hyperparameters for TLDR across different backbone models.

Unlikelihood Training Hyperparameters			
	Qwen3-8B	**Llama-3.1-8B**	**Gemma-3-4B**
LoRA Rank (r)	4	4	4
LoRA α	8	8	8
LoRA Dropout	0.1	0.1	0.1
GPU		$4 \times$ RTX A6000	
Batch Size (per device)	4	4	4
Gradient Accumulation	2	2	2
Learning Rate	1×10^{-4}	1×10^{-4}	1×10^{-4}
LoRA Start Layer	0	0	0
LoRA End Layer	35	31	33

Table 8: Training hyperparameters for Token-Level Unlikelihood Training across different backbone models.

TDPO Training Hyperparameters			
	Qwen3-8B	**Llama-3.1-8B**	**Gemma-3-4B**
LoRA Rank (r)	4	4	4
LoRA α	8	8	8
LoRA Dropout	0.1	0.1	0.1
GPU		$2 \times$ RTX A6000	
Batch Size (per device)	4	4	4
Gradient Accumulation	4	4	4
Learning Rate	5×10^{-6}	5×10^{-6}	5×10^{-6}
LoRA Start Layer	0	0	0
LoRA End Layer	35	31	33

Table 9: Training hyperparameters for TDPO Training across different backbone models.

A.4 Detailed Results on In-Distribution and Out-of-Distribution Benchmarks

This section provides detailed numerical results for all models and methods under both in-distribution (ID) and out-of-distribution (OOD) settings. Tables 10 and 11 report the corresponding results.

Method	Qwen3-8B	LLaMA-3.1-8B	Gemma-3-4B
TOPL (Ours)	**0.8687 ± 0.0198**	**0.8710 ± 0.0070**	**0.8119 ± 0.0202**
SFT	0.8431 ± 0.0030	0.8436 ± 0.0030	0.8368 ± 0.0081
TLDR	0.8364 ± 0.0119	0.8761 ± 0.0134	0.7829 ± 0.0405
TDPO	0.8304 ± 0.0019	0.8536 ± 0.0014	0.8299 ± 0.0024
Unlikelihood	0.8375 ± 0.0024	0.8419 ± 0.0017	0.8342 ± 0.0025
SOPL	0.8116	0.7932	0.7359
DPO	0.8069	0.7957	0.7231

Table 10: In-distribution factuality scores across backbone models.

Method	Qwen3-8B	LLaMA-3.1-8B	Gemma-3-4B
TOPL (Ours)	**0.8726 ± 0.0085**	**0.8619 ± 0.0009**	**0.8398 ± 0.0078**
SFT	0.8511 ± 0.0035	0.8555 ± 0.0003	0.8221 ± 0.0043
TLDR	0.8533 ± 0.0132	0.8510 ± 0.0022	0.8327 ± 0.0090
TDPO	0.8057 ± 0.0003	0.8542 ± 0.0050	0.7887 ± 0.0024
Unlikelihood	0.8370 ± 0.0101	0.8658 ± 0.0007	0.8300 ± 0.0007
SOPL	0.8434	0.8143	0.8160
DPO	0.8468	0.8084	0.8127

Table 11: Out-of-distribution factuality scores averaged across the 11 AggreFact datasets.

A.5 Label Sensitivity Dataset Statistics

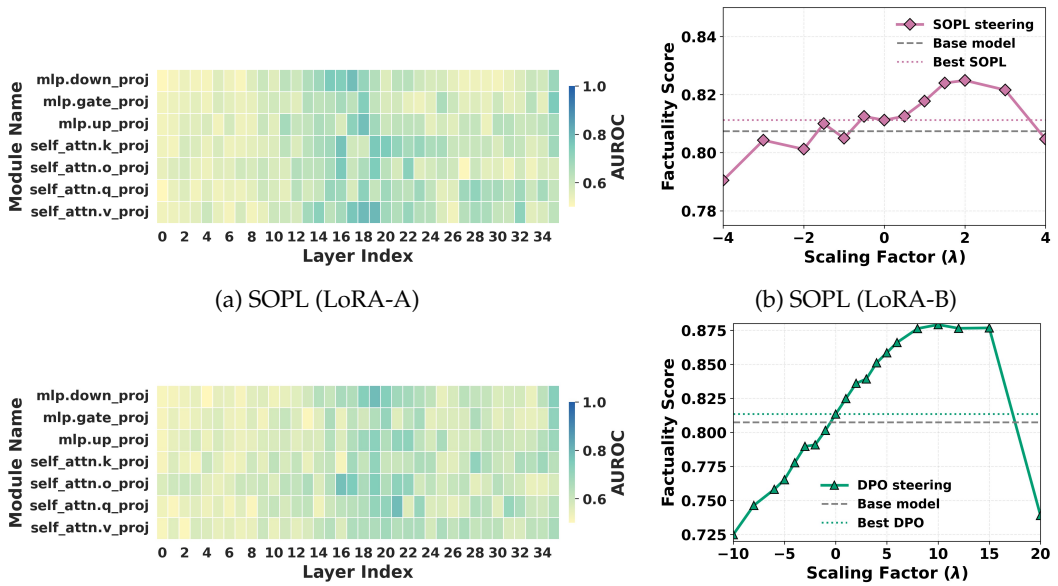
We report the statistics of the datasets used in the label sensitivity analysis, including the number of supervised tokens and the distribution of positive and negative labels for each setting in Table 12.

Setting	#Tokens	% Good	% Bad
Original	2.91M	67.2	32.8
Random	2.91M	49.8	50.2
Balanced	1.40M	49.5	50.5
Bad-skewed	1.43M	32.9	67.1
Good-skewed	1.38M	65.8	34.2
Good-only	1.95M	100.0	0.0
Dense Balanced	1.82M	50.0	50.0

Table 12: Token-level supervision statistics for label sensitivity settings. We report the total number of supervised tokens and the proportion of positive (factual) and negative (hallucinated) labels for each configuration.

A.6 LoRA-A and LoRA-B Analysis for DPO and SOPL

In addition to TOPL and SFT, we further analyze the LoRA-A and LoRA-B components learned by DPO and SOPL. Figure 9 presents the corresponding LoRA-A separability heatmaps and LoRA-B steering curves.



(a) SOPL (LoRA-A)

(b) SOPL (LoRA-B)

(c) DPO (LoRA-A)

(d) DPO (LoRA-B)

Figure 9: LoRA-A and LoRA-B analysis for DPO and SOPL. **Top row:** SOPL; **bottom row:** DPO. **Left column:** LoRA-A separability heatmaps. **Right column:** LoRA-B steering curves under different scaling factors λ. While both methods exhibit relatively weak LoRA-A separability compared to TOPL, they still demonstrate clear LoRA-B steering behavior.

Compared to TOPL, both DPO and SOPL exhibit substantially weaker LoRA-A separability across layers and modules, suggesting that their learned low-rank subspaces provide less discriminative structure between factual and non-factual tokens. Interestingly, however, both methods still demonstrate clear LoRA-B steering effects under scaling.

This observation suggests that the LoRA-B steering effect is not unique to TOPL; instead, the primary distinction appears to lie in the structure of LoRA-A. A possible explanation is that effective steering behavior depends not only on learning useful steering directions through LoRA-B, but also on learning appropriate conditions under which these directions should be activated. In particular, LoRA-A appears to determine how selectively the learned steering directions are applied within the hidden representation space. Although DPO and SOPL

learn usable LoRA-B directions, their weaker LoRA-A structure may limit how effectively these directions are utilized after merging. By contrast, the stronger LoRA-A separability learned by TOPL may allow the model to more consistently activate the appropriate steering directions, leading to improved factuality and stronger OOD robustness.

A.7 Detailed Machine Translation Results

Table 13 reports the complete machine translation results on Qwen3-8B under both in-distribution (ID) and out-of-distribution (OOD) settings. Higher values indicate better translation quality.

Method	ID	OOD
SFT	0.8843	0.9192
TLDR	0.8860	0.9226
Unlikelihood	0.8606	0.8972
TOPL (Ours)	**0.8947**	**0.9300**

Table 13: Detailed Machine translation results on Qwen3-8B. TOPL achieves the best performance in both ID and OOD settings.