DRIFT: Difference-Aware Reinforcement Through Iterative Fine-Tuning for Language Models

Wenjie Liao[1], **Xiaohui Song**[1], **Haonan Lu**[1]

[1]Guangdong OPPO Mobile Telecommunications Corp., Ltd.
liaowenjie3040@gmail.com, songxiaohui@oppo.com, luhaonan@oppo.com

Abstract

Self-play fine-tuning has emerged as a promising approach to improve Large Language Models (LLMs) without additional human annotations. However, existing methods struggle with complex generation tasks requiring long context understanding, where models produce partially correct outputs interleaved with errors. Traditional approaches train on entire sequences uniformly, failing to distinguish between well-predicted and erroneous regions, leading to diluted learning signals and slow convergence. We propose DRIFT (Difference-aware Reinforcement through Iterative Fine-Tuning), a novel self-play framework that selectively trains on prediction differences. DRIFT introduces two key innovations: (1) Difference-Aware Masking (DAM) that identifies and masks common subsequences between model outputs and ground truth, focusing training exclusively on error regions; (2) Occurrence-Aware Loss (OAL) that provides position-invariant vocabulary supervision, complementing the position-sensitive adversarial loss. This dual mechanism enables models to correct both positional and lexical errors effectively. Theoretically, we prove that DRIFT converges when masked distributions align. Empirically, we evaluate DRIFT on diverse summarization benchmarks using Qwen2.5-3B and LLaMA-3.1-8B models. Results show that DRIFT significantly outperforms both supervised fine-tuning (SFT) and self-play fine-tuning (SPIN), achieving up to 16% improvement on SAMSum dialogue summarization tasks while maintaining general capabilities. Notably, DRIFT breaks the performance ceiling of continued SFT and demonstrates superior efficiency compared to holistic self-play methods, validating that targeted optimization on prediction differences is crucial for structured text generation tasks.

Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities across diverse tasks, from complex reasoning to creative content generation (Anil et al. 2023; Albert Q. Jiang, Chaplot et al. 2023; Grattafiori et al. 2024; DeepSeek-AI et al. 2025; OpenAI 2025). A critical challenge in deploying these models lies in aligning them with human values and intentions, ensuring they produce helpful, harmless, and honest outputs (Ouyang et al. 2022). While Reinforcement Learning from Human Feedback (RLHF) has emerged as the dominant paradigm for this alignment (Christiano et al. 2017; Azar et al. 2024; Xiong et al. 2023; Azar et al. 2024), it requires expensive human preference data and complex multi-stage training procedures.

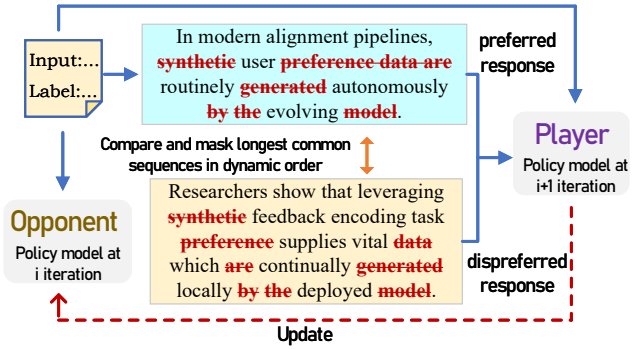

Figure 1: Overview of the DRIFT training framework. At iteration i, the opponent model generates synthetic responses, which are compared with ground-truth labels to extract their LCS. DRIFT masks these shared subsequences to focus optimization on differing regions. The updated model then becomes the next opponent, enabling iterative self-improvement.

Recent advances have explored self-play mechanisms (e.g.,(Dubois et al. 2025; Wu et al. 2024; Chen et al. 2024; Cheng et al. 2024; Fang et al. 2025)) as a promising alternative for LLM alignment without additional human annotations.Self-Play Fine-Tuning (SPIN) (Chen et al. 2024) demonstrates that LLMs can be iteratively improved by training against their previous versions, where the model learns to distinguish between human-generated responses and those from its earlier iterations. This approach effectively converts weak models to strong ones through a two-player game framework (Silver et al. 2017; Goodfellow et al. 2014), achieving performance comparable to preference-based methods without requiring preference data.

However, existing self-play methods face notable challenges when applied to complex text generation (Peng et al. 2023; Tan et al. 2024).In such tasks, models often produce outputs that are partially correct, yet structurally disordered or lexically inconsistent.These responses may contain seg-

ments interleaved with hallucinations, severe repetition, and quality degradation (Que et al. 2024; Pham, Sun, and Iyyer 2024), which are less problematic in short-form tasks but detrimental in long-form settings where coherence and precision are paramount. Traditional self-play approaches treat the entire sequence uniformly during training, failing to distinguish between well-predicted spans and erroneous regions. Furthermore, in extended sequences, models exhibit two distinct error patterns: local coherence failures (where semantically appropriate tokens appear in wrong positions) and global vocabulary drift (where the model selects increasingly inappropriate tokens as generation progresses). These indiscriminate optimization not only dilutes the learning signal but also slows convergence and hampers the model's ability to make targeted improvements.

In this work, we introduce DRIFT (Difference-aware Reinforcement through Iterative Fine-Tuning), a novel self-play framework that addresses these limitations through targeted learning mechanisms.Our key insight is that effective self-improvement should focus exclusively on the model's prediction differences rather than reinforcing already-correct patterns. DRIFT achieves this through two complementary innovations: (1) we propose a Difference-Aware Masking (DAM) mechanism that identifies common longest subsequences (LCS) between the reference model's outputs and ground truth labels, then masks these shared portions during training. This ensures that the adversarial loss concentrates solely on regions where the model makes errors. (2) we introduce an Occurrence-Aware Loss (OAL) that complements the position-sensitive adversarial training. Unlike traditional sequence-level losses, OAL evaluates the maximum probability of each ground truth token across all positions in the generated sequence, providing position-invariant supervision. This design captures vocabulary-level correctness independent of positional accuracy, addressing a critical gap in existing self-play methods. Our contributions are threefold:

- We identify and analyze the unique challenges of self-play fine-tuning for complex text generation tasks in LLMs, revealing how holistic sequence treatment and undifferentiated error types limit extended output performance.
- We propose DRIFT, a difference-aware self-play framework combining targeted adversarial training via LCS-based masking with position-invariant vocabulary supervision for effective structured generation.
- Extensive experiments on diverse benchmarks demonstrate that the proposed DRIFT significantly outperforms supervised fine-tuning (SFT) and baseline self-play method (SPIN) on summarization task, achieving better alignment quality without any additional human-annotated preference data.

Related Work

RLHF with Preference Model. Reinforcement Learning from Human Feedback (RLHF), initially proposed by Christiano et al. (Christiano et al. 2017), introduces a framework in which a reward model is first trained to approximate human preferences, followed by the application of reinforcement learning algorithms to optimize policy performance with respect to this learned reward. Building upon this foundation, Ouyang et al. (Ouyang et al. 2022) applied RLHF to fine-tune instruction-following large language models, a process that led to the development of ChatGPT and sparked widespread interest in alignment techniques for LLMs.

The core of most RLHF approaches lies in the Bradley-Terry (BT) model (Bradley and Terry 1952), which assumes that human preferences can be captured through scalar scores assigned by a reward model to responses. More recently, a significant advancement came with Direct Preference Optimization (DPO) by Rafailov et al. (Rafailov et al. 2023), which eliminates the need for explicit reward modeling by deriving a closed-form solution under the BT assumption. In a similar spirit, Zhao et al. (Zhao et al. 2023) proposed Sequence Likelihood Calibration (SLiC) which introduces margin-based calibration to ensure adequate separation between preferred and dispreferred responses. Ethayarajh et al. (Ethayarajh et al. 2024a) developed Kahneman-Tversky Optimization (KTO) based on prospect theory, offering a more nuanced modeling of human utility functions. Hong, Lee, and Thorne (Hong, Lee, and Thorne 2024) introduced Odds Ratio Preference Optimization (ORPO), which unifies supervised fine-tuning and preference alignment in a single training phase, eliminating the need for maintaining a separate reference policy.

Recognizing that human preferences often violate transitivity assumptions in score-based models, recent research explores broader preference frameworks.Azar et al. (Azar et al. 2023) proposed Identity Preference Optimization (IPO), which directly optimizes the preference probability without assuming an underlying score function. Munos et al. (Munos et al. 2024) reformulated RLHF as a two-player constant-sum game, deriving robust Nash equilibrium policies inherently suited to handling non-transitive preferences with rigorous theoretical guarantees. Similarly, Wang, Liu, and Jin (Wang, Liu, and Jin 2023) extended this framework to multi-step decision processes, reducing the preference learning problem to solving zero-sum Markov games.

Self-Play Fine-Tuning. Self-play, first introduced by Samue (Samuel 1959) and later advanced by Tesauro et al. (Tesauro et al. 1995), refers to a learning paradigm in which an agent improves its policy by repeatedly interacting with versions of itself. This approach has received considerable attention, particularly in the context of multi-agent reinforcement learning (MARL), due to its ability to progressively increase the difficulty and diversity of training scenarios without requiring external supervision. A landmark achievement in this domain is AlphaGo Zero (Silver et al. 2017), which attained superhuman performance in the game of Go solely through self-play, without access to human data. Building upon this foundation, subsequent studies have proposed various extensions and implementations of self-play in both competitive and cooperative settings (Arjovsky, Chintala, and Bottou 2017; Lanctot et al. 2017; Bansal et al. 2017; Hernandez-Leal, Kartal, and Taylor 2018; Muller et al. 2019; Singh et al. 2024), further val-

idating its potential as a scalable and autonomous learning mechanism. Recently, the self-play paradigm has been successfully adapted to language model post-training, offering a promising alternative to traditional RLHF approaches. Unlike single-round optimization procedures common in preference-based methods (Rafailov et al. 2023; Zhao et al. 2023; Azar et al. 2023; Ethayarajh et al. 2024b), self-play fine-tuning operates iteratively: each round generates new training data using the policy from the previous round, creating a continuous improvement cycle without requiring additional human annotations.

Self-play fine-tuning has demonstrated applicability in both settings—with or without access to human preference data. For instance, Singh et al. (Singh et al. 2024) introduced an Expectation-Maximization (EM) based framework in which, at each iteration, new responses are generated and assigned reward scores, and the policy is updated by fine-tuning on samples with high rewards. Chen et al. (Chen et al. 2024) proposed a supervised self-play approach (SPIN) wherein preference pairs are synthetically constructed by designating policy-generated responses as inferior and human-generated responses as superior, followed by applying Direct Preference Optimization (DPO) to train a new policy at each iteration. In a related direction, Yuan et al. (Yuan et al. 2025) presented the Self-Rewarding Language Models framework, in which the model autonomously evaluates and assigns preferences to its own outputs. The annotated data are then used to iteratively fine-tune the model using DPO. Collectively, these studies highlight the effectiveness of iterative fine-tuning strategies in enhancing model performance, even in the absence of explicit human preference signals.

Problem Setting and Preliminaries

We consider a Large Language Model (LLM), denoted by p_θ, where θ represents its trainable parameters. Given an input prompt $x = [x_1, x_2, ..., x_n]$, the model aims to produce a response $y = [y_1, y_2, ..., y_m]$ by drawing samples from the conditional distribution $p_\theta(\cdot \mid x)$. Each element x_i or y_i corresponds to a token from a predefined vocabulary. The LLM is structured as an auto-regressive generator, predicting the next token based solely on the prior tokens already generated. Consequently, the response distribution follows a Markov structure, and can be factorized sequentially as:

$$p_\theta(y \mid x) = \prod_{j=1}^{m} p_\theta(y_i \mid x, y_{<j}) \tag{1}$$

where $y_{<j} = [y_1, y_2, ..., y_{j-1}]$ for $j \geq 2$, and $y_{<1}$ is empty. This formulation highlights that the generation process is conditioned on both the input prompt and the previously produced tokens.

Self-Play Fine-Tuning (SPIN)

Self-Play Fine-Tuning (SPIN) (Chen et al. 2024) represents a paradigm shift in LLM alignment by eliminating the need for human preference data through an iterative self-improvement mechanism. The method draws inspiration from two key concepts: the adversarial training framework of GANs (Goodfellow et al. 2014) and the iterative nature of self-play in game theory (Wu et al. 2024; Gao et al. 2024; Azar et al. 2023). SPIN formulates fine-tuning as a two-player game where the LLM plays against its previous iterations. At iteration t, the opponent player (previous model π_{θ_t}) generates synthetic responses y' for prompts x from the SFT dataset. The main player (current model $\pi_{\theta_{t+1}}$) learns to distinguish between these synthetic responses and human-generated responses y from the original dataset. The training objective at iteration $t+1$ is formulated as:

$$\mathcal{L}_{SPIN}(\theta_{t+1}, \theta_t) = \mathrm{E}_{x,y,y'}[\ell(\beta \mathbf{log} \frac{\pi_{\theta_{t+1}}(y \mid x)}{\pi_{\theta_t}(y \mid x)} - \beta \mathbf{log} \frac{\pi_{\theta_{t+1}}(y' \mid x)}{\pi_{\theta_t}(y' \mid x)})] \tag{2}$$

where ℓ represents the logistic loss function $\ell(t) = \mathbf{log}(1 + \mathbf{exp}(-t))$. The SPIN objective bears structural similarity to DPO but with a crucial difference: While DPO requires paired preference data (y_w, y_l) annotated by humans, SPIN automatically constructs preference pairs by treating human responses as preferred and self-generated responses as disprefered. This self-supervised approach enables iterative improvement without external supervision.

While SPIN demonstrates impressive empirical results, it treats entire sequences uniformly during training. In structured text generation, where model outputs may be partially correct, this holistic approach fails to focus learning on the specific regions where the model errs. This observation motivates our difference-aware enhancement in DRIFT, which identifies and selectively trains on prediction differences to achieve more efficient learning.

Methodology

In this section, we introduce DRIFT (Difference-aware Reinforcement through Iterative Fine-Tuning), a novel self-play fine-tuning method that enhances LLM performance on text generation tasks without requiring additional human or AI feedback. Consider a high-quality supervised fine-tuning (SFT) dataset $S_{\mathbf{SFT}} = \{(x_i, y_i)\}_{i=1}^n$, sampled from the prompt distribution $q(x)$ and human response distribution $p_{\mathbf{data}}(y \mid x)$. Given a supervised fine-tuned LLM π_θ, further application of standard SFT approach in (2) becomes ineffective, particularly for structured text generation where the model produces partially correct but suboptimal responses. Furthermore, obtaining a suitable dataset of preference annotations for reinforcement learning-based fine-tuning approaches, such as DPO, becomes impractical in the absence of human evaluators or external AI-generated feedback. To address this challenge, the Self-Play Fine-Tuning (SPIN) strategy (Chen et al. 2024) is introduced, utilizing artificial data produced by the LLM itself.

However, we observe that in extended sequences, the quality gap between human responses y and model-generated responses $y' \sim \pi_\theta(\cdot \mid x)$ is not uniformly distributed but concentrated in specific regions where the model makes systematic errors. This insight motivates our

difference-aware approach: rather than training on entire sequences, we identify and focus on the regions where the model genuinely needs improvement, while introducing complementary mechanisms to address both positional and vocabulary-level errors inherent in structured text generation. Fig. 1 illustrates the main part of DRIFT framework.

DRIFT: Difference-aware Reinforcement through Iterative Fine-Tuning

Following the self-play paradigm, we formulate DRIFT as a two-player game where the main player (current model) learns to distinguish between human and machine-generated responses, while the opponent (previous iteration) generates responses that challenge this discrimination. The key innovation lies in our selective training mechanism that identifies and leverages prediction differences. At iteration $t+1$, the opponent π_{θ_t} generates pseudo labels y' for prompt x in the SFT dataset. Our method consists of three core components: (1) identifying differences through LCS-based masking, (2) training with difference-aware adversarial loss and occurrence-aware loss, and (3) updating to the next iteration.

Difference-Aware Masking (DAM) Mechanism. Given a human response $y = [y_1, y_2, ..., y_m]$ and opponent-generated response $y' = [y'_1, y'_2, ..., y'_{m'}]$, we first compute their Longest Common Subsequence (LCS):

$$\mathbf{LCS}(y, y') = \mathbf{argmax}_{s \in S} |s| \tag{3}$$

where S is the set of all common subsequences between y and y'. The LCS identifies tokens that appear in the same dynamic order in both sequences, representing the model's correct predictions. We then create masked versions by removing the LCS as illustrated in Fig. 1:

$$\tilde{y} = y \backslash \mathbf{LCS}(y, y') \tag{4}$$

$$\tilde{y}' = y' \backslash \mathbf{LCS}(y, y') \tag{5}$$

This masking ensures that the adversarial loss focuses exclusively on regions where the model makes errors, dramatically improving training efficiency for long sequences where large portions may be correctly generated.

Adversarial Training with Masked Sequences. Following the integral probability metric (IPM) framework (Müller 1997), we formulate our objective function to train the main player that can effectively distinguish between masked human responses and masked synthetic responses. The main player f_{t+1} at iteration $t+1$ maximizes the expected value gap between the target data distribution and the opponent player's distribution:

$$f_{t+1} = \arg\max_{f \in \mathcal{F}_t} \mathrm{E}_{x,y,y'}[f(x, \tilde{y}) - f(x, \tilde{y}')] \tag{6}$$

where the expectation is computed over $x \sim q(\cdot)$, $y \sim p_{data}(\cdot \mid x)$, $y' \sim \pi_{\theta_t}(\cdot \mid x)$, and $\tilde{y}, \tilde{y}'$ denote the masked sequences obtained through our DAM mechanism. The function class $\mathcal{F}_t$ depends on the opponent distribution π_{θ_t}. The value $f_{t+1}(x, \tilde{y})$ represents the main player's confidence that the masked sequence $\tilde{y}$ originates from the human distribution rather than the model distribution. To ensure stable optimization and prevent unbounded objectives, we reformulate this as a more general optimization problem as suggested in (Chen et al. 2024):

$$f_{t+1} = \arg\max_{f \in \mathcal{F}_t} \mathrm{E}_{x,y,y'}[\ell(f(x, \tilde{y}) - f(x, \tilde{y}'))] \tag{7}$$

where $\ell(\cdot)$ is a monotonically decreasing convex loss function. This formulation allows for various loss choices while maintaining the core objective of maximizing the discrimination gap. Given the optimized main player f_{t+1}, we now determine how to update the opponent player's parameters. The opponent seeks to generate responses that are indistinguishable from human responses according to the main player's evaluation. This is achieved by finding a policy that maximizes:

$$\mathrm{E}_{x \sim q(\cdot), y \sim p(\cdot|x)}[f_{t+1}(x, \tilde{y})] \tag{8}$$

To prevent excessive deviation from the previous iteration and ensure stable self-play, we incorporate a Kullback-Leibler (KL) regularization term (Kullback and Leibler 1951):

$$\begin{aligned} &\arg\max_{p} \mathrm{E}_{x \sim q(\cdot), \tilde{y} \sim p(\cdot|x)}[f_{t+1}(x, \tilde{y})] \\ &-\beta \mathrm{E}_{x \sim q(\cdot)}[\mathbf{KL}(p(\cdot \mid x) \parallel \pi_{\theta_t}(\cdot \mid x))] \end{aligned} \tag{9}$$

where $\beta > 0$ is the regularization parameter and (9) has a closed-form solution:

$$\hat{p}(\tilde{y} \mid x) \propto \pi_{\theta_t}(\tilde{y}, x) \exp(\frac{1}{\beta} f_{t+1}(x, \tilde{y})) \tag{10}$$

To ensure that this optimal policy can be realized by LLM parameterization, we require $\hat{p}(\tilde{y} \mid x) = \pi_\theta(\tilde{y} \mid x)$ from some $0 \in \Theta$. This constraint suggests following function class for f_{t+1}:

$$\mathcal{F}_t = \left\{ \beta \cdot \log \frac{\pi_\theta(\tilde{y} \mid x)}{\pi_{\theta_t}(\tilde{y} \mid x)} \mid \theta \in \Theta \right\} \tag{11}$$

Substituting this function class into (7), we can obtain the adversarial training objective for DAM Mechanism in DRIFT:

$$\begin{aligned} \mathcal{L}_{\mathbf{DAM}}(\theta, \theta_t) = \mathrm{E}_{x,y,y'}[\ell(&\beta \log \frac{\pi_\theta(\tilde{y} \mid x)}{\pi_{\theta_t}(\tilde{y} \mid x)} \\ &-\beta \log \frac{\pi_\theta(\tilde{y}' \mid x)}{\pi_{\theta_t}(\tilde{y}' \mid x)})] \end{aligned} \tag{12}$$

For the loss function ℓ, we adopt the logistic loss $\ell(t) = \log(1 + \exp(-t))$ as suggested in (Chen et al. 2024).This formulation elegantly captures the adversarial dynamics: the model learns to assign higher probability to masked human sequences while lowering probability on its own masked generations, with the masking ensuring focus on regions of genuine improvement rather than already-correct predictions.

Occurrence-Aware Loss (OAL). To complement the position-sensitive adversarial loss, we introduce OAL that evaluates vocabulary-level correctness independent of position. For each token v in the vocabulary $\mathcal{V}$ that appears in the human response y, we compute:

Algorithm 1: DRIFT (Difference-aware Reinforcement through Iterative Fine-Tuning)

Require: SFT dataset $\{(x_i, y_i)\}_{i\in[N]}$, initial model π_{θ_0}, iterations T

for $t = 0, \ldots, T-1$ **do**
  for $i = 1, \ldots, N$ **do**
    Generate synthetic response $y_i' \sim \pi_{\theta_t}(\cdot|x_i)$
    Compute $\mathbf{LCS}(y_i, y_i')$
    Create masked sequences $\tilde{y}_i, \tilde{y}_i'$
  end for
  Update $\theta_{t+1} = \arg\min_\theta \mathcal{L}_{\mathbf{DRIFT}}(\theta, \theta_t)$
end for
Output: θ_T

$$p_{\max}(v \mid x) = \max_{j\in[1,m']} \pi_\theta(y_j' = v \mid x, y_{<j}') \tag{13}$$

where m' is the number of token generated by policy model, $p_{\max}(v \mid x)$ represents the maximal probability of token v along all the positions of sequence generated by policy model π_{θ_t}. The OAL is then defined as:

$$\mathcal{L}_{\mathbf{OAL}}(\theta) = -\mathrm{E}_{x,y}[\sum_{v\in\mathcal{V}(y)} \log p_{\max}(v \mid x)] \tag{14}$$

where $\mathcal{V}(y)$ denotes the set of unique token in y. This loss ensures that the model assigns high probability to ground-truth tokens somewhere in its output, providing flexibility in generation while maintaining vocabulary accuracy.

End-to-end Training Objective. The final DRIFT objective combines both losses:

$$\mathcal{L}_{\mathbf{DRIFT}}(\theta, \theta_t) = \mathcal{L}_{\mathbf{DAM}}(\theta, \theta_t) + \lambda \mathcal{L}_{\mathbf{OAL}}(\theta) \tag{15}$$

where $\lambda > 0$ balances the two objectives. This dual mechanism ensures comprehensive learning: DAM corrects position-dependent errors in differentiating regions, while OAL maintains global vocabulary coherence.

Following the self-play protocol, after optimizing θ_{t+1} using the combined objective, the opponent is updated for the next iteration:

$$\theta_{t+1} = \arg\min_{\theta\in\Theta} \mathcal{L}_{\mathbf{DRIFT}}(\theta, \theta_t) \tag{16}$$

The new model $\pi_{\theta_{t+1}}$ then becomes the opponent for iteration $t+2$, creating an iterative improvement cycle. The detailed algorithm is presented in Algorithm 1.

Theorem 4.1 (Convergence of DRIFT). Under mild assumptions on the loss function ℓ (monotonicity and convexity), DRIFT converges to a stationary point where the masked distributions align:

$$\pi_{\theta^*}(\tilde{y}' \mid x) = p_{data}(\tilde{y} \mid x, y' \sim \pi_{\theta^*}) \tag{17}$$

Proof Sketch. The proof follows from analyzing the fixed-point conditions of the combined objective. At convergence, the gradient of $\mathcal{L}_{\mathbf{DRIFT}}$ vanishes, implying that the model can no longer distinguish between masked human and self-generated responses.

Theorem 4.2 (Efficiency of Difference-Aware Training). Let $\gamma = \mathrm{E}[\mid \mathbf{LCS}(y, y') \mid / \mid y \mid]$ be the average fraction of correctly predicted tokens. The gradient variance of DRIFT is reduced by a factor of $(1-\gamma)^2$ compared to SPIN:

$$\mathbf{Var}[\nabla_\theta \mathcal{L}_{\mathbf{DRIFT}}] \leq (1-\gamma)^2 \cdot \mathbf{Var}[\nabla_\theta \mathcal{L}_{\mathbf{SPIN}}] \tag{18}$$

This reduction in gradient variance leads to more stable training and faster convergence, particularly beneficial for long sequences where γ is typically large. The theoretical properties and algorithmic design of DRIFT ensure effective learning for complex text generation, addressing the unique challenges that arise when self-play methods are applied to extended sequences.

Experiments

This section provides a comprehensive empirical analysis of DRIFT on summarization tasks. Our findings highlight several key insights: (1) DRIFT markedly enhances model generation performance across multiple summarization datasets by breaking the limit of SFT; (2) while achieving significant gains in generation quality, DRIFT maintains or even slightly improves general capabilities on standard benchmarks; (3) compared to SPIN, DRIFT achieves superior performance on generation tasks, validating the effectiveness of our targeted training mechanism for structured outputs; (4) iterative training is a necessary component in DRIFT as it breaks the limit of multi-epoch training, with consistent improvements observed across multiple iterations.

Experiment Setup

Model and Training Datasets We evaluate DRIFT on two state-of-the-art instruction-tuned language models: Qwen2.5-3B-Instruct (Qwen et al. 2025) and Llama3.1-8B-Instruct (Grattafiori et al. 2024),representing different scales and architectures to assess the generalizability of our approach. Our training corpus comprises diverse summarization datasets across languages and domains: Chinese datasets including LCSTS (Hu, Chen, and Zhu 2015) (20k samples) and CNewsSum (Wang et al. 2021) (20k samples); English datasets including XSum (Narayan, Cohen, and Lapata 2018) (20k samples) and SAMSum (Gliwa et al. 2019) (15k samples); and one proprietary conversation summarization datasets - Chunk (7k). This results in a total training corpus of 82k samples, providing diverse challenges in terms of language, domain, and summarization style. At each iteration, we use the current model to generate synthetic summaries for all training prompts, then apply DRIFT's difference-aware training mechanism.

Evaluation We conduct evaluation using two complementary approaches. For summarization performance, we evaluate on testing set of corresponding 5 summarization tasks (2k samples per task), reporting ROUGE scores (including ROUGE-1, ROUGE-2, and ROUGE-L) (Chin-Yew 2004) as our metric. For general capabilities, we follow SPIN's evaluation protocol using the HuggingFace Open LLM Leaderboard (Beeching et al. 2023), which comprehensively evaluates models across five tasks: GSM8K (Cobbe et al. 2021)

Models	Iteration Methods	LCSTS	CNewsSum	XSum	SAMSum	Chunk	ROUGE-1 Ave	ROUGE-2 Ave	ROUGE-L Ave
Qwen-2.5 -3B-Instruct	SFT epoch 1	31.62	43.33	37.23	47.47	60.36	44.00	24.90	37.62
	SFT epoch 2	30.95	43.34	37.24	46.66	60.36	43.71(-0.29)	24.66(-0.24)	37.33(-0.29)
	DRIFT iteration 1	33.70	45.00	39.52	**54.64**	63.60	**47.29(+3.29)**	26.31(+1.41)	39.78(+2.16)
	DRIFT iteration 2	34.62	46.10	39.97	56.93	64.13	**48.35(+1.06)**	26.66(+0.35)	39.95(+0.17)
	DRIFT iteration 3	34.76	46.66	40.54	57.08	64.46	**48.60(+0.25)**	26.74(+0.08)	40.63(+0.68)
	DRIFT iteration 4	34.68	46.57	39.96	57.95	64.51	**48.74(+0.14)**	26.88(+0.14)	40.80(+0.17)
LLaMA-3.1 -8B-Instruct	SFT epoch 1	30.31	43.94	41.01	49.93	60.54	45.15	27.03	37.47
	SFT epoch 2	30.13	43.67	40.56	48.41	60.35	44.60(-0.55)	25.50(-1.53)	38.16(+0.69)
	DRIFT iteration 1	35.57	47.34	44.51	**65.94**	64.68	**51.61(+6.46)**	28.27(+1.24)	**42.41(+4.94)**
	DRIFT iteration 2	36.88	49.56	46.90	69.98	61.73	**53.01(+1.40)**	29.25(+0.98)	44.07(+1.66)
	DRIFT iteration 3	37.52	50.90	48.92	67.81	66.80	**54.24(+1.24)**	30.21(+0.96)	**47.36(+3.29)**
	DRIFT iteration 4	38.85	50.98	50.35	68.66	66.34	**55.04(+0.80)**	30.42(+0.21)	47.52(+0.16)

Table 1: Performance comparison of DRIFT across summarization tasks

Models	Iteration Methods	GSM8K	Hellaswag	WinoGrande	MMLU	TruthfulQA	Leaderboard Ave
Qwen-2.5 -3B-Instruct	SFT epoch 1	71.87	74.54	61.01	65.42	34.52	61.47
	SFT epoch 2	71.57	74.78	62.12	65.33	37.21	62.20
	DRIFT iteration 1	74.45	75.20	62.84	65.52	37.33	**63.07(+1.60)**
	DRIFT iteration 2	74.91	75.34	62.68	65.42	36.35	62.94
	DRIFT iteration 3	73.46	74.97	61.97	65.46	36.47	62.47
	DRIFT iteration 4	74.60	74.56	63.15	65.36	35.25	62.58
LLaMA-3.1 -8B-Instruct	SFT epoch 1	81.35	60.31	61.49	66.31	32.93	60.48
	SFT epoch 2	81.77	56.41	60.70	64.24	34.64	59.55
	DRIFT iteration 1	81.38	61.54	61.41	65.75	42.23	**62.46(+1.98)**
	DRIFT iteration 2	81.90	61.63	62.28	65.46	43.33	62.92
	DRIFT iteration 3	80.91	61.53	61.89	65.15	44.55	62.81
	DRIFT iteration 4	81.44	59.42	59.98	64.80	43.70	61.89

Table 2: Performance comparison of DRIFT across general benchmarks

for mathematical reasoning; HellaSwag (Zellers et al. 2019), and Winogrande (Sakaguchi et al. 2021) for commonsense reasoning; MMLU (Hendrycks et al. 2021) for multi-task understanding; and TruthfulQA (Lin, Hilton, and Evans 2022) for truthfulness. All evaluations employ the standardized Language Model Evaluation Harness (Gao et al. 2021). We leave further implementation details to Appendix B with training and evaluation setting.

DRIFT Effectively Enhances Structured Text Generation Performance

We evaluate the effectiveness of DRIFT by comparing it against standard SFT baselines across our diverse summarization benchmarks. Table 1 presents the results of ROUGE-1 scores, the average ROUGE-2, and the average ROUGE-L scores (detailed results are recorded in Appendix B) for both Qwen2.5-3B-Instruct and LLaMA-3.1-8B-Instruct models. A critical observation is the performance degradation when continuing SFT beyond the first epoch. For Qwen2.5-3B-Instruct, SFT epoch 2 shows consistent deterioration across all datasets, with scores dropping by 0.67 points on LCSTS, and notably 0.81 points on SAMSum. This degradation is even more pronounced for LLaMA-3.1-8B-Instruct, where scores decrease by 0.45 points on XSum and substantial drops of 1.52 points on SAMSum. This phenomenon confirms that naive continued training on the same SFT data leads to overfitting and performance plateau.

In contrast,starting from the SFT epoch 1 checkpoint, DRIFT iteration 1 achieves substantial improvements across all summarization tasks. For Qwen2.5-3B-Instruct, the average ROUGE-1 score increased by 3.29 and SAMSum shows a dramatic increase of 7.17 points (from 47.47 to 54.64). For LLaMA-3.1-8B-Instruct, the enhancements are more pronounced under DRIFT training. The model achieves remarkable gains of 6.46 points on average ROUGE-1 score and 4.94 points on average ROUGE-L score, while SAMSum increases by 16.01 points (from 49.93 to 65.94).Importantly, these generation improvements do not come at the cost of general capabilities. The Leaderboard Average scores (Detailed performances are presented in Table 2) show that DRIFT maintains or even slightly improves general performance. Subsequent iterations continue this trend of incremental improvement across various tasks. For instance, LLama3.1-8B-Instruct shows additional gains from iteration 1 to iteration 2 (+1.40) on average ROUGE-1, though the

improvement at iteration t+1 is naturally smaller (+1.24 and +0.80). This convergence pattern aligns with our theoretical analysis and demonstrates the self-limiting nature of the DRIFT optimization process.

Comparison with SPIN To directly compare DRIFT with SPIN, we conduct controlled experiments where both methods start from the same Qwen2.5-3B-Instruct checkpoint and use identical training data. Figure 2 shows the ROUGE-1 performance trajectory over multiple iterations. DRIFT demonstrates clear superiority over SPIN throughout the training process. While both methods improve upon the SFT baseline, DRIFT exhibits a steeper learning curve and consistently maintains a significant performance gap across all iterations. This advantage validates our hypothesis that focusing on prediction differences rather than training on entire sequences leads to more effective learning in complex text generation tasks. The detailed task-wise comparisons are provided in Table 4 in Appendix B.

Ablation Studies

In this subsection, we conduct ablation studies to examine the contribution of each component in DRIFT and investigate the necessity of iterative training. Our analysis focuses on the impact of our key innovations and training dynamics. All the experiments are conducted upon SFT 1 epoch baseline of Qwen2.5-3B-Instruct and detailed experimental results are recorded in Appendix B.

Component Analysis To understand the contribution of each proposed component, we conduct ablation experiments by removing the Difference-Aware Masking (DAM) mechanism and Occurrence-Aware Loss (OAL) separately. Results show that both components are crucial for DRIFT's performance. Removing the DAM mechanism leads to a drop in average summarization ROUGE-1 score from 47.29 to 46.49, demonstrating that focusing on prediction differences is essential for effective learning. Without OAL, performance decreases from 47.29 to 46.99, indicating that position-invariant vocabulary supervision provides valuable complementary learning signals. The more significant impact of removing DAM validates our core hypothesis that selective training on erroneous regions is critical for structured text generation tasks. Detailed results are presented in Table 5 in Appendix B.

Iterative Training vs. Training for More Epochs We investigate whether the performance gains from iterative training can be achieved by simply training for more epochs without updating the opponent model. After completing iteration 1, we continue training for three additional epochs while keeping the synthetic data fixed. The results in Figure 3 clearly demonstrate the superiority of iterative training. Although extended training in one iteration brings marginal gains, it cannot match the performance from proper iteration updates. This plateau effect confirms that updating the opponent model is essential for continued progress, as static synthetic data becomes less informative once the model has learned to distinguish it from human responses. The performance trajectory exhibits diminishing returns across epochs,

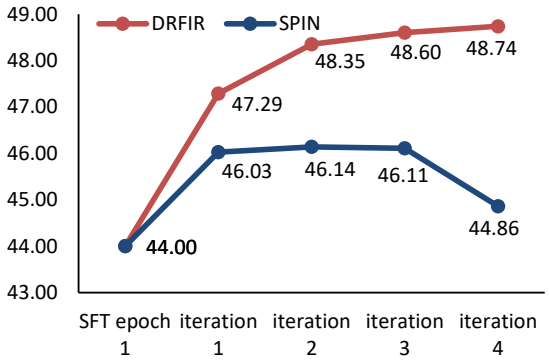

Figure 2: Performance comparison between DRIFT and SPIN across iterations on summarization tasks using Qwen2.5-3B-Instruct.

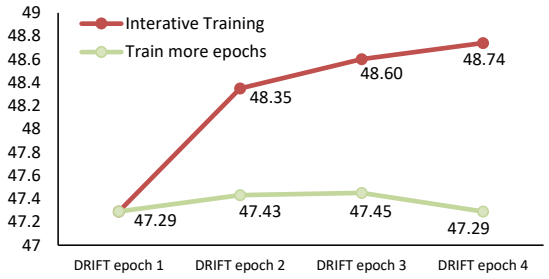

Figure 3: Performance comparison of DRIFT between Iterative Training and Training for More Epochs on summarization tasks using Qwen2.5-3B-Instruct.

whereas initiating a new iteration with updated synthetic data yields substantial improvements.

Conclusion

This paper introduces DRIFT, a novel self-play fine-tuning method that converts weak language models to strong ones by addressing the unique challenges of structural generation tasks requiring long context understanding. Central to DRIFT is a difference-aware mechanism that identifies and selectively trains on prediction errors while preserving correct patterns. Through the synergy of Difference-Aware Masking (DAM) and Occurrence-Aware Loss (OAL), DRIFT enables language models to iteratively refine their generation capabilities without requiring additional human annotations or preference data. Our empirical results demonstrate that DRIFT significantly outperforms both standard SFT and SPIN across diverse summarization benchmarks. The focused training on prediction differences proves particularly effective for dialogue summarization, while maintaining or even improving general capabilities. This validates our hypothesis that selective optimization on erroneous regions leads to more efficient learning than holistic sequence training. Future work could extend the difference-aware paradigm to other complex generation tasks such as code synthesis and mathematical reasoning.

References

Albert Q. Jiang, A. M. C. B., Alexandre Sablayrolles; Chaplot, D. S.; et al. 2023. Mistral 7B. arXiv:2310.06825.

Anil, R.; Dai, A. M.; Firat, O.; Johnson, M.; Lepikhin, D.; Passos, A.; et al. 2023. PaLM 2 Technical Report. arXiv:2305.10403.

Arjovsky, M.; Chintala, S.; and Bottou, L. 2017. Wasserstein generative adversarial networks. In *International conference on machine learning*, 214–223. PMLR.

Azar, M. G.; Guo, Z. D.; Piot, B.; Munos, R.; Rowland, M.; Valko, M.; and Calandriello, D. 2024. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics*, 4447–4455. PMLR.

Azar, M. G.; Rowland, M.; Piot, B.; Guo, D.; Calandriello, D.; Valko, M.; and Munos, R. 2023. A General Theoretical Paradigm to Understand Learning from Human Preferences. arXiv:2310.12036.

Bansal, T.; Pachocki, J.; Sidor, S.; Sutskever, I.; and Mordatch, I. 2017. Emergent complexity via multi-agent competition. *arXiv preprint arXiv:1710.03748*.

Beeching, E.; Fourrier, C.; Habib, N.; Han, S.; Lambert, N.; Rajani, N.; Sanseviero, O.; Tunstall, L.; and Wolf, T. 2023. Open LLM Leaderboard. https://huggingface.co/spaces/open-llm-leaderboard. Accessed on July 30, 2025.

Bradley, R. A.; and Terry, M. E. 1952. Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika*, 39(3/4): 324–345.

Chen, Z.; Deng, Y.; Yuan, H.; Ji, K.; and Gu, Q. 2024. Self-play fine-tuning converts weak language models to strong language models. *arXiv preprint arXiv:2401.01335*.

Cheng, J.; Liu, X.; Wang, C.; Gu, X.; Lu, Y.; Zhang, D.; Dong, Y.; Tang, J.; Wang, H.; and Huang, M. 2024. SPaR: Self-Play with Tree-Search Refinement to Improve Instruction-Following in Large Language Models. *arXiv preprint arXiv:2412.11605*.

Chin-Yew, L. 2004. Rouge: A package for automatic evaluation of summaries. In *Proceedings of the Workshop on Text Summarization Branches Out, 2004*.

Christiano, P. F.; Leike, J.; Brown, T.; Martic, M.; Legg, S.; and Amodei, D. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.

Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; Hesse, C.; and Schulman, J. 2021. Training Verifiers to Solve Math Word Problems. arXiv:2110.14168.

DeepSeek-AI; Guo, D.; Yang, D.; Zhang, H.; Song, J.; Zhang, R.; Xu, R.; Zhu, Q.; et al. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. arXiv:2501.12948.

Dubois, Y.; Galambosi, B.; Liang, P.; and Hashimoto, T. B. 2025. Length-Controlled AlpacaEval: A Simple Way to Debias Automatic Evaluators. arXiv:2404.04475.

Ethayarajh, K.; Xu, W.; Muennighoff, N.; Jurafsky, D.; and Kiela, D. 2024a. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*.

Ethayarajh, K.; Xu, W.; Muennighoff, N.; Jurafsky, D.; and Kiela, D. 2024b. KTO: Model Alignment as Prospect Theoretic Optimization. arXiv:2402.01306.

Fang, W.; Liu, S.; Zhou, Y.; Zhang, K.; Zheng, T.; Chen, K.; Song, M.; and Tao, D. 2025. SeRL: Self-Play Reinforcement Learning for Large Language Models with Limited Data. arXiv:2505.20347.

Gao, L.; Tow, J.; Biderman, S.; Black, S.; DiPofi, A.; Foster, C.; Golding, L.; Hsu, J.; McDonell, K.; Muennighoff, N.; et al. 2021. A framework for few-shot language model evaluation. *Version v0. 0.1. Sept*, 10: 8–9.

Gao, Z.; Chang, J. D.; Zhan, W.; Oertell, O.; Swamy, G.; Brantley, K.; Joachims, T.; Bagnell, J. A.; Lee, J. D.; and Sun, W. 2024. REBEL: Reinforcement Learning via Regressing Relative Rewards. arXiv:2404.16767.

Gliwa, B.; Mochol, I.; Biesek, M.; and Wawer, A. 2019. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. *arXiv preprint arXiv:1911.12237*.

Goodfellow, I. J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27.

Grattafiori, A.; Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; et al. 2024. The Llama 3 Herd of Models. arXiv:2407.21783.

Hendrycks, D.; Burns, C.; Basart, S.; Zou, A.; Mazeika, M.; Song, D.; and Steinhardt, J. 2021. Measuring Massive Multitask Language Understanding. arXiv:2009.03300.

Hernandez-Leal, P.; Kartal, B.; and Taylor, M. E. 2018. Is multiagent deep reinforcement learning the answer or the question? A brief survey. *learning*, 21: 22.

Hong, J.; Lee, N.; and Thorne, J. 2024. ORPO: Monolithic Preference Optimization without Reference Model. arXiv:2403.07691.

Hu, B.; Chen, Q.; and Zhu, F. 2015. Lcsts: A large scale chinese short text summarization dataset. *arXiv preprint arXiv:1506.05865*.

Kullback, S.; and Leibler, R. A. 1951. On information and sufficiency. *The annals of mathematical statistics*, 22(1): 79–86.

Lanctot, M.; Zambaldi, V.; Gruslys, A.; Lazaridou, A.; Tuyls, K.; Pérolat, J.; Silver, D.; and Graepel, T. 2017. A unified game-theoretic approach to multiagent reinforcement learning. *Advances in neural information processing systems*, 30.

Lin, S.; Hilton, J.; and Evans, O. 2022. TruthfulQA: Measuring How Models Mimic Human Falsehoods. arXiv:2109.07958.

Müller, A. 1997. Integral probability metrics and their generating classes of functions. *Advances in applied probability*, 29(2): 429–443.

Muller, P.; Omidshafiei, S.; Rowland, M.; Tuyls, K.; Perolat, J.; Liu, S.; Hennes, D.; Marris, L.; Lanctot, M.; Hughes, E.; et al. 2019. A generalized training approach for multiagent learning. *arXiv preprint arXiv:1909.12823*.

Munos, R.; Valko, M.; Calandriello, D.; Azar, M. G.; Rowland, M.; Guo, Z. D.; Tang, Y.; Geist, M.; Mesnard, T.; Michi, A.; Selvi, M.; Girgin, S.; Momchev, N.; Bachem, O.; Mankowitz, D. J.; Precup, D.; and Piot, B. 2024. Nash Learning from Human Feedback. arXiv:2312.00886.

Narayan, S.; Cohen, S. B.; and Lapata, M. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *arXiv preprint arXiv:1808.08745*.

OpenAI. 2025. Introducing openai o3 and o4-mini, 2025. https://openai.com/index/introducing-o3-and-o4-mini/. Accessed: 2025-07-08.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744.

Peng, B.; Quesnelle, J.; Fan, H.; and Shippole, E. 2023. YaRN: Efficient Context Window Extension of Large Language Models. arXiv:2309.00071.

Pham, C. M.; Sun, S.; and Iyyer, M. 2024. Suri: Multi-constraint Instruction Following for Long-form Text Generation. arXiv:2406.19371.

Que, H.; Duan, F.; He, L.; Mou, Y.; Zhou, W.; Liu, J.; Rong, W.; Wang, Z. M.; Yang, J.; Zhang, G.; Peng, J.; Zhang, Z.; Zhang, S.; and Chen, K. 2024. HelloBench: Evaluating Long Text Generation Capabilities of Large Language Models. arXiv:2409.16191.

Qwen; :; Yang, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; et al. 2025. Qwen2.5 Technical Report. arXiv:2412.15115.

Rafailov, R.; Sharma, A.; Mitchell, E.; Manning, C. D.; Ermon, S.; and Finn, C. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36: 53728–53741.

Sakaguchi, K.; Bras, R. L.; Bhagavatula, C.; and Choi, Y. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9): 99–106.

Samuel, A. L. 1959. Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, 3(3): 210–229.

Silver, D.; Schrittwieser, J.; Simonyan, K.; Antonoglou, I.; Huang, A.; Guez, A.; Hubert, T.; Baker, L.; Lai, M.; Bolton, A.; et al. 2017. Mastering the game of go without human knowledge. *nature*, 550(7676): 354–359.

Singh, A.; Co-Reyes, J. D.; Agarwal, R.; Anand, A.; Patil, P.; Garcia, X.; Liu, P. J.; et al. 2024. Beyond Human Data: Scaling Self-Training for Problem-Solving with Language Models. arXiv:2312.06585.

Tan, H.; Guo, Z.; Shi, Z.; Xu, L.; Liu, Z.; Feng, Y.; Li, X.; Wang, Y.; Shang, L.; Liu, Q.; and Song, L. 2024. PROXYQA: An Alternative Framework for Evaluating Long-Form Text Generation with Large Language Models. arXiv:2401.15042.

Tesauro, G.; et al. 1995. Temporal difference learning and TD-Gammon. *Communications of the ACM*, 38(3): 58–68.

Wang, D.; Chen, J.; Wu, X.; Zhou, H.; and Li, L. 2021. CNewSum: a large-scale Chinese news summarization dataset with human-annotated adequacy and deducibility level. *arXiv preprint arXiv:2110.10874*.

Wang, Y.; Liu, Q.; and Jin, C. 2023. Is rlhf more difficult than standard rl? a theoretical perspective. *Advances in Neural Information Processing Systems*, 36: 76006–76032.

Wu, Y.; Sun, Z.; Yuan, H.; Ji, K.; Yang, Y.; and Gu, Q. 2024. Self-Play Preference Optimization for Language Model Alignment. arXiv:2405.00675.

Xiong, W.; Dong, H.; Ye, C.; Wang, Z.; Zhong, H.; Ji, H.; Jiang, N.; and Zhang, T. 2023. Iterative preference learning from human feedback: Bridging theory and practice for rlhf under kl-constraint. *arXiv preprint arXiv:2312.11456*.

Yuan, W.; Pang, R. Y.; Cho, K.; Li, X.; Sukhbaatar, S.; Xu, J.; and Weston, J. 2025. Self-Rewarding Language Models. arXiv:2401.10020.

Zellers, R.; Holtzman, A.; Bisk, Y.; Farhadi, A.; and Choi, Y. 2019. HellaSwag: Can a Machine Really Finish Your Sentence? arXiv:1905.07830.

Zhao, Y.; Joshi, R.; Liu, T.; Khalman, M.; Saleh, M.; and Liu, P. J. 2023. Slic-hf: Sequence likelihood calibration with human feedback. *arXiv preprint arXiv:2305.10425*.