W2S-AlignTree: Weak-to-Strong Inference-Time Alignment for Large Language Models via Monte Carlo Tree Search

Zhenyu Ding[1], Yuhao Wang[2], Tengyue Xiao[1], Haoying Wang[1], Guojun Ma[3] *, Mingyang Wan[1], Caigui Jiang[1], Ning Ding[1] *

[1]State Key Laboratory of Human-Machine Hybrid Augmented Intelligence, National Engineering Research Center for Visual Information and Applications, and Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an, China
[2]School of Computer Science and Technology, Xi'an Jiaotong University, Xi'an, China
[3]Tsinghua Shenzhen International Graduate School, Tsinghua University, Shenzhen, China
{dzyxjtu, wyhao, 748984521, whywhy}@stu.xjtu.edu.cn, {magjhaha, wanmingyang}@126.com, {cgjiang, ding.ning}@xjtu.edu.cn

Abstract

Large Language Models (LLMs) demonstrate impressive capabilities, yet their outputs often suffer from misalignment with human preferences due to the inadequacy of weak supervision and a lack of fine-grained control. Training-time alignment methods like Reinforcement Learning from Human Feedback (RLHF) face prohibitive costs in expert supervision and inherent scalability limitations, offering limited dynamic control during inference. Consequently, there is an urgent need for scalable and adaptable alignment mechanisms. To address this, we propose W2S-AlignTree, a pioneering plug-and-play inference-time alignment framework that synergistically combines Monte Carlo Tree Search (MCTS) with the Weak-to-Strong Generalization paradigm for the first time. W2S-AlignTree formulates LLM alignment as an optimal heuristic search problem within a generative search tree. By leveraging weak model's real-time, step-level signals as alignment proxies and introducing an Entropy-Aware exploration mechanism, W2S-AlignTree enables fine-grained guidance during strong model's generation without modifying its parameters. The approach dynamically balances exploration and exploitation in high-dimensional generation search trees. Experiments across controlled sentiment generation, summarization, and instruction-following show that W2S-AlignTree consistently outperforms strong baselines. Notably, W2S-AlignTree raises the performance of `Llama3-8B` from 1.89 to 2.19, a relative improvement of 15.9% on the summarization task.

Code — https://github.com/alexzdy/W2S-AlignTree

Introduction

Large Language Models (LLMs) have demonstrated capabilities in natural language understanding, text generation, and complex reasoning that approach or even surpass human performance. However, a crucial issue is the misalignment between their behaviors and human values, frequently manifesting as biased, harmful, or false content (Weidinger et al. 2022). Both academia and industry have increasingly focused on developing LLM alignment methods to ensure helpful, honest, and harmless responses. The prevailing alignment paradigm is Reinforcement Learning from Human Feedback (RLHF), which typically involves supervised fine-tuning (SFT), reward model training (Christiano et al. 2017) and Proximal Policy Optimization (PPO) (Ouyang et al. 2022). As shown in Fig. 1, despite its empirical successes, RLHF faces several critical challenges. First, RLHF relies heavily on large-scale, high-quality human-annotated data to train reward models or reinforcement learning algorithms, which are known to be unstable and computationally expensive. Parameter-efficient fine-tuning methods like LoRA (Hu et al. 2022) mitigate compute costs by freezing base model weights, but they consequently constrain generative flexibility. Additionally, Direct Preference Optimization (DPO) (Rafailov et al. 2023) and its variants (Zhou et al. 2023; Meng, Xia, and Chen 2024) reframe preference learning as contrastive loss minimization, eliminating explicit reward modeling and RL sampling to improve training stability and efficiency. However, DPO and RLHF both face the same problem: they rely on sequence-level and post-hoc

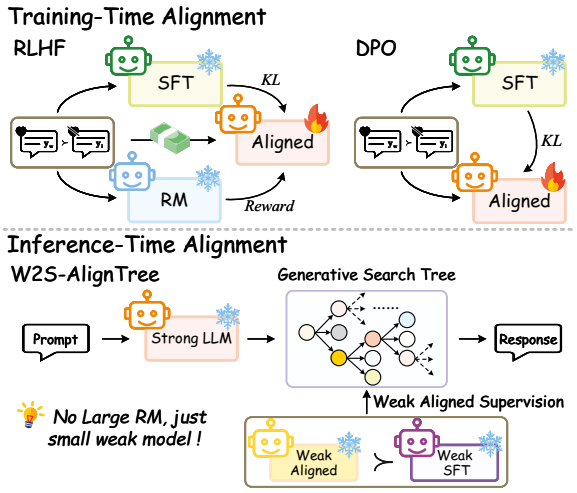


Figure 1: W2S-AlignTree vs. Training-Time Alignment. Unlike RLHF and DPO, W2S-AlignTree enables fine-grained inference-time alignment using weak model signals—without costly reward models or parameter updates.

*Corresponding author.

feedback available only during training, which leaves them unable to provide immediate, fine-grained control at inference time. More fundamentally, as LLMs grow in scale, their behaviors may exceed the capacity of human annotation or other limited supervision, making feedback signals inadequate for aligning them (Leike, Sutskever, and OpenAI 2023). Early attempts like OpenAI's Weak-to-Strong Generalization (W2SG) (Burns et al. 2023) seek to use weaker supervision to align stronger models, yet remain largely preliminary and offer no real-time control for inference.

With the emergence of frontier LLMs (OpenAI 2024; Guo et al. 2025), Inference-time Scaling has become a promising paradigm. It enhances model performance by strategically allocating additional computing resources during the model inference phase, offering a new avenue to overcome the optimization bottleneck in the training phase. Methods such as Chain-of-Thought (CoT) (Wei et al. 2022) and Tree-of-Thought (ToT) (Yao et al. 2023) have significantly enhanced LLM performance on complex reasoning by guiding multi-step or parallel thinking. Building on this trend, inference-time alignment methods, including CBS (Zhou et al. 2024) and TPO (Li et al. 2025), have also appeared. While these methods guide model outputs via preference signals without parameter updates, they often struggle to explore complex generative spaces or enable fine-grained control.

Monte Carlo Tree Search (MCTS) (Silver et al. 2016, 2017) addresses these limitations by employing the Upper Confidence bounds applied to Trees (UCT) (Kocsis and Szepesvári 2006) to balance node visits and returns in large search spaces, offering a promising approach to improve LLM inference. While MCTS has already been successfully applied to enhance mathematical reasoning (Zhang et al. 2024a; Qi et al. 2024) and task planning (Zhang et al. 2024b; Zhai et al. 2025), its potential for inference-time alignment remains largely unexplored. Even in alignment-related work MCTS-DPO (Xie et al. 2024), MCTS is primarily used for offline data generation rather than real-time model guidance. Critically, within the W2SG framework, leveraging MCTS to dynamically align strong models under weak supervision represents an underexplored key issue for future research.

To bridge the gap between powerful LLMs and effective alignment under weak supervision, this paper proposes **W2S-AlignTree**, the first framework that integrates MCTS with the W2SG paradigm. As indeicated in Fig. 1, we formulate preference alignment as a search process over a generative search tree, where a weak model provides dynamic guidance to efficiently and scalably explore the strong model's response space.

The main contributions of this paper are as follows:

- We introduce **W2S-AlignTree**, a plug-and-play MCTS-based alignment framework built on a "weak guidance-strong exploration" mechanism. By injecting preference proxy signals from a weak LLM, it enables dynamic and fine-grained control over strong LLMs with unified step-level guidance and sequence-level evaluation.
- We design an Entropy-Aware Prioritized UCT (EA-PUCT) selection rule that integrates policy entropy and prior probabilities to adaptively capture uncertainty, reducing premature convergence and improving trajectory diversity and quality in complex alignment tasks.
- Comprehensive experiments show W2S-AlignTree significantly boosts LLM alignment on a broad spectrum of challenging tasks, including controlled-sentiment generation, summarization, and instruction-following.

Preliminaries & Problem

Preliminaries

RLHF. Assume the input prompt $\mathbf{x}$ from probability distribution $p(\mathbf{x})$, and a complete model response $\mathbf{y}$. The model to be aligned is denoted as $\pi(\mathbf{y}|\mathbf{x})$, while the reference model is denoted as $\pi_{\text{ref}}(\mathbf{y}|\mathbf{x})$ (e.g., SFT model). A reward function $r(\mathbf{x}, \mathbf{y})$ measures the quality of the responses. The objective of RLHF can be written as:

$$\begin{aligned} \arg\max_{\pi} \quad & \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x}),\, \mathbf{y}\sim\pi(\mathbf{y}|\mathbf{x})}\big[r(\mathbf{x},\mathbf{y})\big] \\ \text{s.t.} \quad & \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}\Big[\mathbb{D}_{\text{KL}}\big(\pi(\mathbf{y}|\mathbf{x}) \,\big\|\, \pi_{\text{ref}}(\mathbf{y}|\mathbf{x})\big)\Big] \le \epsilon, \end{aligned} \tag{1}$$

where $\mathbb{D}_{\text{KL}}$ restricts the optimized model π from deviating too much from the reference (unaligned) model π_{ref}.

For this constrained optimization, a globally optimal closed-form solution $\pi^*(\mathbf{y}|\mathbf{x})$ exists, and its relationship with the reward function can be expressed via the Lagrangian formulation (Ouyang et al. 2022):

$$r(\mathbf{x},\mathbf{y}) = \beta \log \frac{\pi^*(\mathbf{y}|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}|\mathbf{x})} + \beta \log Z(\mathbf{x}), \tag{2}$$

where $Z(\mathbf{x}) = \sum_{\mathbf{y}} \pi_{\text{ref}}(\mathbf{y}|\mathbf{x}) \exp\left(\frac{1}{\beta} r(\mathbf{x},\mathbf{y})\right)$ denotes the partition function. This term acts as a normalization constant, ensuring that $\pi^*(\mathbf{y}|\mathbf{x})$ forms a valid distribution by summing to 1 over all possible responses $\mathbf{y}$. β controls the KL regularization and implicitly scales the reward.

DPO. DPO (Rafailov et al. 2023) streamlines RLHF by using its closed-form solution to cast the alignment objective as a Bradley–Terry contrastive learning problem (Bradley and Terry 1952), thereby eliminating reward model training and RL sampling to improve differentiability and stability. Given a prompt $\mathbf{x}$ with corresponding accepted and rejected responses $\mathbf{y}_{\text{w}}$ and $\mathbf{y}_{\text{l}}$, it optimizes the objective about π:

$$\begin{aligned} &\mathcal{L}_{\text{DPO}}(\pi;\pi_{\text{ref}}) = -\mathbb{E}_{(\mathbf{x},\mathbf{y}_{\text{w}},\mathbf{y}_{\text{l}})\sim\mathcal{D}} \\ &\left[\log\sigma\left(\beta\log\frac{\pi(\mathbf{y}_{\text{w}}|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}_{\text{w}}|\mathbf{x})} - \beta\log\frac{\pi(\mathbf{y}_{\text{l}}|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}_{\text{l}}|\mathbf{x})}\right)\right], \end{aligned} \tag{3}$$

where $\mathcal{D}$ represents the preference dataset, and $\sigma(\cdot)$ is the sigmoid function. DPO regards $\beta \log \frac{\pi(\mathbf{y}|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}|\mathbf{x})}$ as an implicit reward, where $Z(\mathbf{x})$ is naturally canceled in the objective.

Problem Formulation

Despite the theoretical elegance and practical successes of RLHF and DPO, they both typically rely on sparse, sequence-level rewards, which are only available after full responses are generated. This "post-adjustment" paradigm hinders the provision of real-time feedback and fine-grained alignment during inference (Rafailov et al. 2024; Shao et al. 2024). We address this by introducing a value function decomposition method, integrating the structural preference signals for real-time alignment guidance.

Definition 1 (Token-level Reward Decomposition). Consider LLM generation as a token-level Markov Decision Process (MDP): the state at t is $s_t = (\mathbf{x}, \mathbf{y}_{<t})$, and the action is the next token $a_t = y_t$ sampled from the vocabulary. To obtain a dense reward, we decompose the sequence-level alignment reward $r(\mathbf{x}, \mathbf{y})$. Using the closed-form solution of $\pi^*(\mathbf{y}|\mathbf{x})$ and the chain rule, it can be expressed as:

$$r(\mathbf{x}, \mathbf{y}) = \beta \log \frac{\pi^*(\mathbf{y}|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}|\mathbf{x})} = \beta \sum_{t=1}^{|\mathbf{y}|} \log \frac{\pi^*(y_t|\mathbf{x}, \mathbf{y}')}{\pi_{\text{ref}}(y_t|\mathbf{x}, \mathbf{y}')}, \quad (4)$$

where $\mathbf{y}' = \mathbf{y}_{<t}$ represents the prefix up to token $t-1$, $\mathbf{y}_{|\mathbf{y}|}$ is the EOS token, and $Z(\mathbf{x})$ is omitted as it does not influence the objective. This converts sparse sequence-level rewards into a continuous stream of token-level evaluations.

Definition 2 (Intermediate Value Function). We define an intermediate value function $V^*(\mathbf{x}, \mathbf{y}')$ that represents the optimal expected future return for a partial sequence $\mathbf{y}'$ (see Appendix A.3 for details). By the Bellman optimality equations (Puterman 2014), the cumulative log-likelihood ratio in Eq. 4 for $\mathbf{y}'$ relates to $V^*(\mathbf{x}, \mathbf{y}')$:

$$\beta \log \frac{\pi^*(\mathbf{y}'|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}'|\mathbf{x})} = \begin{cases} V^*(\mathbf{x}, \mathbf{y}'), & \text{if } \mathbf{y}' \neq \mathbf{y} \\ r(\mathbf{x}, \mathbf{y}), & \text{if } \mathbf{y}' = \mathbf{y}, \end{cases} \quad (5)$$

where $V^*(\mathbf{x}, \mathbf{y}')$ indicates the promise of a prefix for fully aligned responses, and the absence of rewards at intermediate steps matches sequence-level alignment's sparse-reward settings. In other words, generating from $(\mathbf{x}, \mathbf{y}')$ with high $\beta \log \frac{\pi^*(\mathbf{y}'|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}'|\mathbf{x})}$ has potential to yield a response $\mathbf{y}$ with high overall return $\beta \log \frac{\pi^*(\mathbf{y}|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}|\mathbf{x})}$ (Zhou et al. 2024).

Definition 3 (Weak-to-Strong Proxy Mapping). Considering the prohibitive cost of repeated training, we adopt the W2SG paradigm to achieve scalable alignment at inference time. Our core idea is a proxy mapping that enables an unaligned strong model π_{strong} to "steal" preferences from a pre-aligned weak model π^*_{weak}. During the strong model's inference process, we utilize a prefix-dependent proxy value function $V_{\text{proxy}}(\mathbf{x}, \mathbf{y}')$ based on Definition 2:

$$V_{\text{proxy}}(\mathbf{x}, \mathbf{y}') = \log \frac{\pi^*_{\text{weak}}(\mathbf{y}'|\mathbf{x})}{\pi^{\text{ref}}_{\text{weak}}(\mathbf{y}'|\mathbf{x})}, \quad (6)$$

where β is also ignored since it scales all paths equally. This V_{proxy} provides dense, dynamic and step-by-step feedback rather than a static reward. This granular signal can deeply couple with search-based decoding process. At each generation step, V_{proxy} directly modifies the strong model's sampling probabilities, steering its decoding to the weak model's preference. The precise integration of V_{proxy} with search-based alignment will be detailed in the next section.

Methodology

This section introduces the proposed W2S-AlignTree, which formalizes LLM alignment as an optimal heuristic search problem during the inference phase. W2S-AlignTree uses a dual-stage strategy: first, MCTS expands solution space steered by V_{proxy}; then, the globally optimal leaf node is selected as the final output. To adapt exploration to model uncertainty, we enhance UCT with an Entropy-Aware bonus, allowing dynamic adjustment of exploration–exploitation based on the strong network's local entropy. Please refer to Appendix A.5 for more details.

Overview of W2S-AlignTree

LLM generation requires strict adherence to specific alignment preferences, which can be precisely modeled as an optimal search problem over a **generative search tree**. In this tree, each node $s_t = (\mathbf{x}, \mathbf{y}')$ represents the MDP state with prompt $\mathbf{x}$ and current prefix $\mathbf{y}'$, and each edge corresponds to the action $a_t = y_t$. A full root-to-leaf path forms a complete candidate response $\mathbf{y}$. Given the exponential growth of the generation space, traditional greedy decoding often fails to find best solutions. To address this, W2S-AlignTree aims to steer the generation of the strong model $\pi_{\text{strong}}(y_t|\mathbf{x}, \mathbf{y}')$ at each step. This guidance signal originate not from a costly external reward model but from a proxy value V_{proxy} derived from the weaker model as per Definition 3. Our goal is to select the next token y_t to maximize the following function:

$$\begin{aligned} &\arg\max_{y_t} \mathcal{G}(\mathbf{x}, \mathbf{y}', y_t) \\ &= \arg\max_{y_t} [\log \pi_{\text{strong}}(y_t|\mathbf{x}, \mathbf{y}') + V_{\text{proxy}}(\mathbf{x}, \mathbf{y}' \circ y_t)], \end{aligned} \quad (7)$$

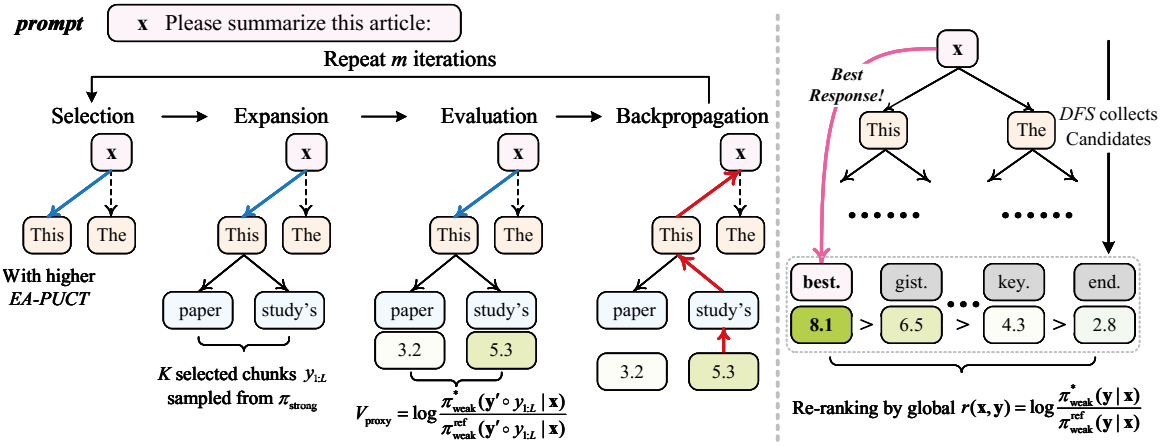
where $s' = \mathbf{y}' \circ y_t$ denotes the newly reached state. The objective offers a principled unification of strong model's inherent generative capabilities with the alignment preferences of the weak model, making it naturally well-suited for the MCTS framework. During the MCTS process, V_{proxy} can be directly considered as the immediate reward $R(s')$ upon reaching the new state. To address the semantic discrepancy of V_{proxy} when evaluating complete versus partial sequences due to Definition 2, W2S-AlignTree employs a dual-stage tree search to achieve inference-time alignment.

Stage 1: Generative Search Tree Construction

The first stage of W2S-AlignTree constructs a generative search tree. As demonstrated in Fig. 2 (a), we incrementally approximate optimal solutions by four iterative phases:

Selection. The phase traverses the existing tree from the root node, repeatedly selecting the child node with the highest potential until an unvisited leaf node is reached for expansion. We employ an EA-PUCT selection rule instead of classic UCT (detailed in the last section), which dynamically adjusts exploration–exploitation to suit the specific characteristics of LLM output distributions.

Expansion. This phase directly utilizes the strong model's pre-trained parameters to generate new child nodes of the selected leaf node. To reflect the strong model's inherent tendencies (corresponding to the first term in $\mathcal{G}$), we first identify the Top-N most probable tokens based on its π_{strong} predicted distribution at the current state s. We then generalize the token-level generation to a more flexible step-level generation, enhancing search efficiency. For diverse exploration, K distinct tokens are randomly selected from these Top-N



(a) Stage 1: Generative Search Tree Construction (b) Stage 2: Best Candidate Decision

Figure 2: Dual-stage alignment process of W2S-AlignTree. (a) MCTS constructs a generative search tree where candidate chunks are proposed by the strong model and evaluated with step-level proxy values from a weak model. (b) Among all explored paths, W2S-AlignTree decides the response by globally re-ranking based on complete sequence-level alignment scores.

candidates to initiate the formation of new chunks. Each selected token then forms a new chunk $y_{1:L}$ of length L, which is subsequently concatenated to the current sequence $\mathbf{y}'$ to yield the new state s'. This design allows for flexible setting of the step size L to adapt different tasks:

- Fine-grained decision-making: When $L = 1$, MCTS performs precise token-by-token decisions, achieving high-accuracy control over alignment at each step.
- High-level branching: When $L > 1$, MCTS expands a short sequence at a time, effectively reducing the tree depth, thereby enabling more efficient and higher-level exploration in high-dimensional spaces.

Evaluation. Each newly generated node s' (representing the partial sequence $\mathbf{y}' \circ y_{1:L}$) undergoes a crucial evaluation by computing the proxy value, serving as its immediate feedback. We promote the proxy value function of Eq. 6:

$$V_{\text{proxy}}(\mathbf{x}, \mathbf{y}' \circ y_{1:L}) = \log \frac{\pi^*_{\text{weak}}(\mathbf{y}' \circ y_{1:L}|\mathbf{x})}{\pi^{\text{ref}}_{\text{weak}}(\mathbf{y}' \circ y_{1:L}|\mathbf{x})}. \quad (8)$$

The effective value of the node is initialized with this V_{proxy}, serving as the immediate reward $R(s')$ for that node in the tree. If a child node meets the stopping condition (e.g., reaching maximum length or generating `EOS` token), its return $R(s')$ is explicitly set to $-\infty$ to prevent its re-selection in subsequent MCTS iterations. These termination nodes will remain candidates for evaluation in Stage 2.

Backpropagation. The $R(s')$ value obtained from the newly simulated node is precisely back-propagated along its path to the root node, with the each ancestor state's visit count incrementally increasing. Crucially, the the parent state's return $R(s_{\text{p}})$ updates to the maximum of all observed child node returns, following the formula:

$$R(s_{\text{p}}) \leftarrow \max(R(s_{\text{c}})). \quad (9)$$

This fundamental design makes MCTS propagate maximum rather than typical average returns, treating LLM alignment as an optimal search for the single highest-reward sequence rather than a long-term average returns like in an adversarial game (Silver et al. 2016). It ensures that MCTS preserves high-value trajectories, effectively prunes sub-optimal branches, and concentrates computational resources on the most promising paths.

Stage 2: Best Candidate Decision

After MCTS iterations, we obtain a generative search tree built with intermediate step-level V_{proxy}, but a final decision still requires global evaluation. Stage 2 aims to identify the best-aligned complete response among all explored nodes by a robust strategy. As shown in Fig. 2 (b), a recursive Depth-First Search (*DFS*) first collects all complete sequences from root to terminal nodes. Among them, we identify candidates whose penultimate nodes have the highest MCTS-evaluated returns, reflecting paths most promising for alignment. For every candidate, its final global alignment We compute the final global alignment score for each candidate according to:

$$r(\mathbf{x}, \mathbf{y}) = \frac{\log \pi^*_{\text{weak}}(\mathbf{y}|\mathbf{x})}{\log \pi^{\text{ref}}_{\text{weak}}(\mathbf{y}|\mathbf{x})}. \quad (10)$$

This score, which matches the full sentence-level alignment objective in Eq. 5, is then used to re-rank the candidates. The candidate with the highest global alignment score will be ultimately decided as the final aligned output. If no terminal node is found during MCTS iterations (e.g., due to insufficient search budget or an extremely difficult problem), we design a fallback mechanism that selects the node with the highest MCTS return, guaranteeing stability and consistent output. Stage 2 integrates both step-level guidance and sequence-level evaluation, leading to a substantial improvement in the reliability, fidelity and quality of the results.

Entropy-Aware PUCT Selection Rule

During the MCTS selection phase, we design an EA-PUCT rule to adaptively balance exploration and exploitation. The

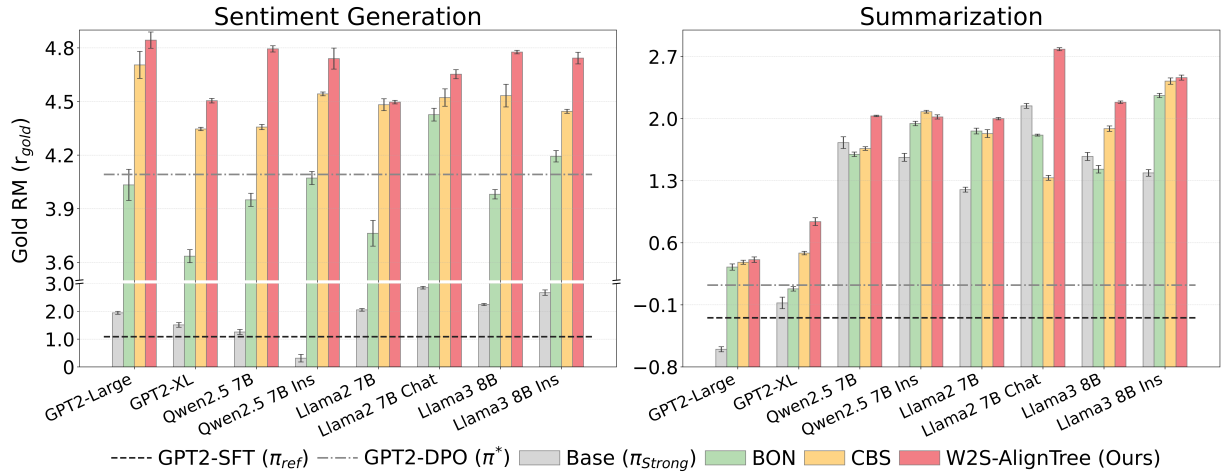


Figure 3: Alignment performance across sentiment generation and summarization. W2S-AlignTree consistently outperforms strong baselines by enabling inference-time alignment with weak model guidance, achieving higher r_{gold} across diverse LLMs. We report mean rewards ($\pm$ standard deviations) across three random seeds. Appendix C.1 provides detailed numerical results.

standard UCT guides exploration by balancing node visit counts and the average value of all historical returns, and PUCT combines prior probabilities from a policy network into UCT. However, the output distribution of π_{strong} often exhibits "peak" effects, causing MCTS to converge prematurely to locally optimal solutions at the expense of diversity and the discovery of better alternatives. To overcome this, we draw on the concept of information entropy into the exploration bonus and propose EA-PUCT, which is defined as:

$$E\text{-}PU(s) = R(s) + c \cdot P(s) \cdot \frac{\sqrt{N(s_{\text{p}})}}{1 + N(s)} \cdot (1 + w \cdot H(s)), \quad (11)$$

where $P(s)$ is the prior probability of the strong model's action leading to state s from the parent s_{p}, $H(s)$ denotes the information entropy, c and w are coefficients. Note that $R(s)$ here denotes the immediate return but not an average, since our task is framed as an optimal search. For a generated chunk of length L, $P(s)$ is precisely defined as the geometric mean of the token-level probabilities: $P(s) = \exp\left(\frac{1}{L}\sum_{t=1}^{L} \log p(y_t | s_{\text{p}}, y_{<t})\right)$. This better penalizes low-probability tokens, boosting robust path exploration.

The information entropy $H(s)$ measures the uncertainty of the strong model's output distribution and is defined as:

$$H(s) = -\sum_{a} P(s, a) \cdot \log P(s, a). \quad (12)$$

By incorporating $(1 + w \cdot H(s))$ as an uncertainty-aware guide for the exploration bonus, EA-PUCT endows the search with information-gain consciousness:

- When the entropy is large, it signals that the model's next-token distribution is highly uncertain; the term $(1 + w \cdot H(s))$ then inflates the exploration bonus, urging MCTS to delve more deeply into diverse trajectories.
- Conversely, a low entropy indicates that the model is confident about the best action; the exploration bonus is suppressed, and the search shifts its emphasis toward exploiting the already-identified high-reward paths.

This mechanism effectively mitigates the premature convergence in MCTS for LLM alignment, markedly enhancing the ability to explore diverse candidate answers in a complex generation space while preventing the excessive randomness that would arise from naive entropy maximization.

Experiments

Experimental Setup

We evaluate W2S-AlignTree's ability to use a weak LLM to guide larger models across diverse tasks, model families and scales. More details are provided in Appendix B.

Task Designs. We evaluate three progressively challenging language tasks using standard datasets: controlled-sentiment generation on `IMDB` (Maas et al. 2011), summarization on `TL;DR` (Stiennon et al. 2020), and instruction-following on `OASST1` (Köpf et al. 2023). For sentiment generation and summarization, we use a supervised fine-tuned `GPT2` model π^*_{weak} (124M parameters) and its DPO-tuned variant $\pi^{\text{ref}}_{\text{weak}}$ to emulate the target behaviors, and jointly use these models to guide a series of larger LLMs. For the more demanding instruction-following, we employ off-the-shelf models and their untuned counterparts (e.g., `Llama-3.2-1B-Instruct` and `Llama-3.2-1B`) as weak guidance signals to steer several powerful LLMs, thereby illustrating that weak models remain universally applicable without any task-specific tuning.

Baselines and Evaluation. We evaluate W2S-AlignTree against several baselines with the same score of Eq. 10 for fair comparison: (1) Base model (π_{strong}): we employ regular generation of the frozen LLMs. (2) Best-of-N (BoN) (Touvron et al. 2023): a post-hoc selection method that chooses the candidate with the highest score from N generated outputs. (3) Chunk-level Beam Search (CBS) (Zhou et al. 2024): an inference-time alignment technique that dynamically integrates reward signals into the beam search process.

	Qwen2.5-7B	Qwen2.5-7B-Instruct	Llama3-8B	Llama3-8B-Instruct	Llama2-7b-hf	Llama2-7b-chat-hf	tulu2-7b	tulu2-7b-dpo
Gold Reward Model: oasst-rm-2-pythia-6.9b Llama3.2-1B-Instruct / Llama3.2-1B: 0.64 / -0.82								
Base	0.90	1.45	-0.68	0.81	-0.75	0.79	-0.13	0.52
BoN	0.91	1.49	-1.28	0.90	-1.22	0.95	0.46	0.57
CBS	0.75	**1.67**	-0.56	**1.13**	-0.52	0.44	-0.73	0.64
W2S-AT (Ours)	**1.33**	1.51	**-0.10**	0.97	**-0.50**	**1.01**	**0.72**	**0.75**
Gold Reward Model: UltraRM-13b Llama3.2-1B-Instruct / Llama3.2-1B: -6.13 / -10.01								
Base	-3.05	0.62	-9.15	-3.45	-9.56	-3.89	-6.48	-4.80
BoN	-1.13	0.57	-10.40	-2.19	-10.52	-3.52	-5.59	-4.24
CBS	-2.21	**1.43**	-9.53	**-1.44**	-8.72	-3.44	-8.04	-4.12
W2S-AT (Ours)	**-1.02**	1.01	**-7.56**	-1.96	**-8.49**	**-3.29**	**-4.37**	**-3.39**

Table 1: Evaluating instruction-following on `OASST1` using W2S-AlignTree (W2S-AT) and representative baselines. We use `Llama3.2-1B-Instruct` and `Llama3.2-1B` as weak guidance models, and both `oasst-rm-2-pythia-6.9b` and `UltraRM-13b` as gold reward models. **Bold** indicates the best r_{gold} score in each column.

Whenever applicable, we contrast inference-time methods with DPO-tuning of the large LLMs, detailed in Appendix C.1. Following common practices (Rafailov et al. 2023; Zhu et al. 2024), we adopt the gold reward-model score r_{gold}, computed by a high-fidelity pre-trained reward model, to assess alignment quality (higher indicates better).

Experimental Results

Experimental results show that W2S-AlignTree consistently and significantly surpasses strong baselines, achieving superior fine-grained alignment across tasks.

Sentiment Generation & Summarization. Results under W2S-AlignTree consistently outperforms strong baselines in both controlled-sentiment generation and summarization, guided by paired `GPT2` models. We mainly evaluate across eight models: in-family `GPT2-Large` (774M) and `GPT2-XL` (1.5B), as well as cross-family mainstream base models such as `Llama2-7b-hf`, `Llama3-8B` and `Qwen2.5-7B`, alongside their aligned versions.

For the controlled-sentiment generation (Fig. 3, Left), W2S-AlignTree consistently elevates the r_{gold} across all models. In particular, it achieves r_{gold} of 4.79 on `Qwen2.5-7B`, representing a significant 10.04% improvement over the second-best CBS with 4.36. It also yields over 5% gains on models such as `Llama3-8B` and `Llama3-8B-Instruct`, highlighting its effectiveness in aligning generated content with target sentiment. In summarization (Fig. 3, Right), all models employing W2S-AlignTree again show robust and stable performance gains except `Qwen2.5-7B-Instruct`, which is also comparable to baselines. For instance, `GPT2-Large` and `GPT2-XL` initially exhibit negative r_{gold} scores of -0.60 and -0.08 due to their weaker inherent capabilities. Under W2S-AlignTree, these scores rise to 0.41 and 0.84, with `GPT2-XL`'s improvement clearly outperforming the baselines and demonstrating enhanced factual and semantic consistency. Moreover, W2S-AlignTree achieved a high r_{gold} of 2.78 on the `Llama-2-7b-chat-hf`, marking a substantial 29.84% improvement compared to the second best direct inference model of 2.14. Generally, CBS greedily aggregates alignment signals at fixed beam width per chunk; while W2S-AlignTree treats signals as explicitly backed-up global values and adaptively explores high-reward long-sequence branches via EA-PUCT, mitigating local-optimum and credit-assignment issues. So W2S-AlignTree enhances alignment across both aligned and unaligned models by leveraging weak model's guidance, ultimately yielding semantically accurate and user-intended outputs.

Instruction-Following. W2S-AlignTree also shows clear advantages in instruction-following across diverse models without task-specific training, highlighting its generality and practical utility. As shown in Tab. 1, we evaluate W2S-AlignTree using `Llama3.2-1B-Instruct` and its untuned `Llama3.2-1B` as weak guidance. To assess more comprehensively, we employ two distinct reward models as gold evaluators: `oasst-rm-2-pythia-6.9b` (Köpf et al. 2023), specifically fine-tuned on `OASST1` to reflect task-specific alignment, and `UltraRM-13b` (Cui et al. 2023), a general-purpose reward model for instruction evaluation across domains. Results indicate that W2S-AlignTree consistently achieves the highest, or occasionally second-highest r_{gold} across most model configurations, demonstrating its stability and strong cross-model generalizability. For instance, when applied to `Qwen2.5-7B`, it raises the score to 1.33, significantly outperforming the next-best method (BoN, 0.91). Under the more stringent UltraRM evaluation, it improves `Llama3-8B`'s score from -9.53 to -7.56. Additional results in Appendix C.2 confirm that W2S-AlignTree remains effective when guided by other smaller models, including but not limited to `Qwen2.5-0.5B`, demonstrating broad adaptability. Overall, W2S-AlignTree provides a robust and scalable approach for inference-time alignment, enabling strong LLMs to generate high-quality, instruction-aligned outputs at low cost.

	Sentiment Gen.		Summarization	
	GPT-XL	Llama3-8B	GPT-XL	Llama3-8B
N-UCT	3.67	3.30	0.67	1.40
RT-UCT	4.09	3.89	0.67	1.63
RT-PUCT	4.39	4.57	0.64	2.12
CMB	3.47	3.29	0.52	1.46
MMB	4.16	4.66	0.61	1.85
W2S-AT	**4.51**	**4.80**	**0.84**	**2.18**

Table 2: Ablation of key components in W2S-AlignTree cross tasks with `Llama3-8B` and `GPT-XL`.

Ablation Studies

We conduct a series of ablation studies to assess the contributions of W2S-AlignTree's key components and hyperparameter settings to alignment performance and robustness.

Key Technical Components. To clarify the impact of each main component of W2S-AlignTree on its alignment, we design five variants for ablation experiments: (1) Naive UCT (N-UCT): basic UCT with average historical return backpropagation. (2) Real-Time UCT (RT-UCT): backpropagating the current return to avoid averaging latency. (3) Real-Time PUCT (RT-PUCT): incorporating strong model priors in selection and real-time return in backpropagation. (4) Child Mean Backpropagation (CMB): backpropagating the mean of all child node returns. (5) Mixed Mean Backpropagation (MMB): backpropagating the average mix of maximum and mean child returns. The comparative experimental results on in-family `GPT2-XL` and cross-family `Llama3-8B` are presented in Tab. 2. The complete W2S-AlignTree, with maximum immediate return backpropagation and EA-PUCT strategy, consistently performs best. Ablation studies reveal that removing key components significantly degrades performance: specifically, the maximum return backpropagation (CMB and MMB) is crucial for preserving optimal paths, while the absence of prior or information entropy (N-UCT, RT-UCT) hinders strategic guidance and adaptive adjustment. RT-PUCT further demonstrates the importance of entropy intervention in preserving the model's exploration capability. These findings validate the effectiveness of W2S-AlignTree's core designs, particularly the key role of maximum return propagation and entropy-enhanced prior guidance in achieving fine-grained alignment.

Hyperparameter Sensitivity. We further analyze the sensitivity of W2S-AlignTree's hyperparameters, mainly focusing on step-level chunk length L and the EA-PUCT exploration constant c here, as visualized in Fig. 4. For L, we observe task-specific preferences. For controlled-sentiment generation, which demands fine-grained controllability, smaller L (e.g., $L = 1$ as token-level decisions) generally yields optimal performance. While for summarization, slightly longer L prove more suitable, emphasizing semantic coherence through wider context capture. Regarding c, we find that model performance is most stable and optimal when $c \in [1.0, 2.0]$, balancing the trade-off between exploration of uncertain paths and exploitation of known high-return branches. Extremely small values of c lead to myopic behavior and early convergence, while larger values introduce unnecessary variance and degrade reliability. And quantitative results in Fig. 4 collectively indicate that W2S-AlignTree consistently achieves high performance under more than one configuration, highlighting its strong stability and robustness. Additional analyses for other parameters are detailed in Appendix C.3.

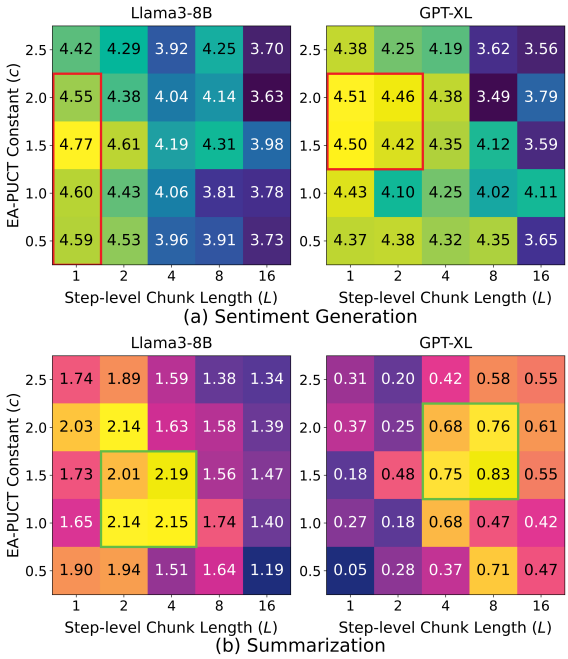


Figure 4: Hyperparameter analysis of W2S-AlignTree to chunk length L and exploration coefficient c across tasks. The areas with better performance are boxed.

Conclusion

We propose **W2S-AlignTree**, a pioneering plug-and-play inference-time alignment framework that overcomes the high costs and limited controllability of training-time alignment. W2S-AlignTree is the first to systematically integrate MCTS with W2S paradigm, recasting LLM alignment as an optimal search problem balancing exploration and exploitation in a generative tree. By leveraging dynamic, step-level signals from a weak model as alignment proxies and introducing the Entropy-Aware PUCT selection rule, W2S-AlignTree achieves fine-grained guidance over the strong model's generation without modifying its parameters. Comprehensive experiments show that W2S-AlignTree consistently enhances alignment quality on tasks such as controlled-sentiment generation, summarization and instruction-following. Superior results across diverse model families, scales, and hyperparameters further highlight the universality and robustness. In conclusion, W2S-AlignTree reveals the effectiveness of integrating MCTS with the W2S paradigm. By enabling more fine-grained and dynamic control over LLM behavior, it offers a scalable alignment solution, opening up a new perspective for practically balancing the safety, controllability and utility of LLMs.

Acknowledgments

This research is supported in part by the National Key Research and Development Program of China (Grant No. 2023ZD0121300), and in part by the National Natural Science Foundation of China (Grant No. 62495092, 62088102).

References

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.

Burns, C.; Izmailov, P.; Kirchner, J. H.; Baker, B.; Gao, L.; Aschenbrenner, L.; Chen, Y.; Ecoffet, A.; Joglekar, M.; Leike, J.; et al. 2023. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision. *arXiv preprint arXiv:2312.09390*.

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.

Cui, G.; Yuan, L.; Ding, N.; Yao, G.; Zhu, W.; Ni, Y.; Xie, G.; Liu, Z.; and Sun, M. 2023. Ultrafeedback: Boosting language models with high-quality feedback. *arXiv preprint arXiv:2310.01377*.

Guo, D.; Yang, D.; Zhang, H.; Song, J.; Zhang, R.; Xu, R.; Zhu, Q.; Ma, S.; Wang, P.; Bi, X.; et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W.; et al. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2): 3.

Kocsis, L.; and Szepesvári, C. 2006. Bandit based monte-carlo planning. In *European conference on machine learning*, 282–293. Springer.

Köpf, A.; Kilcher, Y.; Von Rütte, D.; Anagnostidis, S.; Tam, Z. R.; Stevens, K.; Barhoum, A.; Nguyen, D.; Stanley, O.; Nagyfi, R.; et al. 2023. Openassistant conversations-democratizing large language model alignment. *Advances in neural information processing systems*, 36: 47669–47681.

Leike, J.; Sutskever, I.; and OpenAI. 2023. Introducing Superalignment. https://openai.com/index/introducing-superalignment/. Accessed: 2025-07-17.

Li, Y.; Hu, X.; Qu, X.; Li, L.; and Cheng, Y. 2025. Test-time preference optimization: On-the-fly alignment via iterative textual feedback. *arXiv preprint arXiv:2501.12895*.

Maas, A.; Daly, R. E.; Pham, P. T.; Huang, D.; Ng, A. Y.; and Potts, C. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, 142–150.

Meng, Y.; Xia, M.; and Chen, D. 2024. Simpo: Simple preference optimization with a reference-free reward. *Advances in Neural Information Processing Systems*, 37: 124198–124235.

OpenAI. 2024. Introducing OpenAI o1. https://openai.com/o1/. Accessed: 2024-10-02.

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.

Puterman, M. L. 2014. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

Qi, Z.; Ma, M.; Xu, J.; Zhang, L. L.; Yang, F.; and Yang, M. 2024. Mutual reasoning makes smaller llms stronger problem-solvers. *arXiv preprint arXiv:2408.06195*.

Rafailov, R.; Hejna, J.; Park, R.; and Finn, C. 2024. From r to Q^*: Your language model is secretly a Q-function. *arXiv preprint arXiv:2404.12358*.

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.

Shao, Z.; Wang, P.; Zhu, Q.; Xu, R.; Song, J.; Bi, X.; Zhang, H.; Zhang, M.; Li, Y.; Wu, Y.; et al. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Silver, D.; Huang, A.; Maddison, C. J.; Guez, A.; Sifre, L.; Van Den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. 2016. Mastering the game of Go with deep neural networks and tree search. *nature*, 529(7587): 484–489.

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.

Stiennon, N.; Ouyang, L.; Wu, J.; Ziegler, D.; Lowe, R.; Voss, C.; Radford, A.; Amodei, D.; and Christiano, P. F. 2020. Learning to summarize with human feedback. *Advances in neural information processing systems*, 33: 3008–3021.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q. V.; Zhou, D.; et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35: 24824–24837.

Weidinger, L.; Uesato, J.; Rauh, M.; Griffin, C.; Huang, P.-S.; Mellor, J.; Glaese, A.; Cheng, M.; Balle, B.; Kasirzadeh, A.; et al. 2022. Taxonomy of risks posed by language models. In *Proceedings of the 2022 ACM conference on fairness, accountability, and transparency*, 214–229.

Xie, Y.; Goyal, A.; Zheng, W.; Kan, M.-Y.; Lillicrap, T. P.; Kawaguchi, K.; and Shieh, M. 2024. Monte carlo tree search boosts reasoning via iterative preference learning. *arXiv preprint arXiv:2405.00451*.

Yao, S.; Yu, D.; Zhao, J.; Shafran, I.; Griffiths, T. L.; Cao, Y.; and Narasimhan, K. 2023. Tree of thoughts: Deliberate problem solving with large language models, 2023. *URL https://arxiv. org/abs/2305.10601*, 3: 1.

Zhai, Y.; Yang, T.; Xu, K.; Feng, D.; Yang, C.; Ding, B.; and Wang, H. 2025. Enhancing decision-making for llm agents via step-level q-value models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(25): 27161–27169.

Zhang, D.; Huang, X.; Zhou, D.; Li, Y.; and Ouyang, W. 2024a. Accessing gpt-4 level mathematical olympiad solutions via monte carlo tree self-refine with llama-3 8b. *arXiv preprint arXiv:2406.07394*.

Zhang, D.; Zhoubian, S.; Hu, Z.; Yue, Y.; Dong, Y.; and Tang, J. 2024b. Rest-mcts*: Llm self-training via process reward guided tree search. *Advances in Neural Information Processing Systems*, 37: 64735–64772.

Zhou, Z.; Liu, J.; Shao, J.; Yue, X.; Yang, C.; Ouyang, W.; and Qiao, Y. 2023. Beyond one-preference-fits-all alignment: Multi-objective direct preference optimization. *arXiv preprint arXiv:2310.03708*.

Zhou, Z.; Liu, Z.; Liu, J.; Dong, Z.; Yang, C.; and Qiao, Y. 2024. Weak-to-strong search: Align large language models via searching over small language models. *Advances in Neural Information Processing Systems*, 37: 4819–4851.

Zhu, W.; He, Z.; Wang, X.; Liu, P.; and Wang, R. 2024. Weak-to-strong preference optimization: Stealing reward from weak aligned model. *arXiv preprint arXiv:2410.18640*.