LLMdoctor: Token-Level Flow-Guided Preference Optimization for Efficient Test-Time Alignment of Large Language Models

Tiesunlong Shen[1,2], Rui Mao[1], Jin Wang[2*], Heming Sun[3], Jian Zhang[4], Xuejie Zhang[2], Erik Cambria[1]

[1]Nanyang Technological University
[2]Yunnan University
[3]Yokohama National University
[4]Xi'an Jiaotong University
tensorshen@mail.ynu.edu.cn, rui.mao@ntu.edu.sg, wangjin@ynu.edu.cn, hemingsun@ieee.org, zhangjian062422@stu.xjtu.edu.cn, xjzhang@ynu.edu.cn, cambria@ntu.edu.sg

Abstract

Aligning Large Language Models (LLMs) with human preferences is critical, yet traditional fine-tuning methods are computationally expensive and inflexible. While test-time alignment offers a promising alternative, existing approaches often rely on distorted trajectory-level signals or inefficient sampling, fundamentally capping performance and failing to preserve the generative diversity of the base model. This paper introduces LLMdoctor, a novel framework for efficient test-time alignment that operates via a patient-doctor paradigm. It integrates token-level reward acquisition with token-level flow-guided preference optimization (TFPO) to steer a large, frozen *patient* LLM with a smaller, specialized *doctor* model. Unlike conventional methods that rely on trajectory-level rewards, LLMdoctor first extracts fine-grained, token-level preference signals from the patient model's behavioral variations. These signals then guide the training of the doctor model via TFPO, which establishes flow consistency across all subtrajectories, enabling precise token-by-token alignment while inherently preserving generation diversity. Extensive experiments demonstrate that LLMdoctor significantly outperforms existing test-time alignment methods and even surpasses the performance of full fine-tuning approaches like DPO.

1 Introduction

Large Language Models (LLMs) exhibit impressive capabilities but require careful alignment with human preferences to ensure safe, helpful, and ethical outputs. Traditional alignment approaches like reinforcement learning from human feedback (RLHF) (Ouyang et al. 2022) and direct preference optimization (DPO) (Rafailov et al. 2023) fine-tune LLMs on human preference datasets, incurring substantial computational costs and requiring repeated training to accommodate diverse or evolving user preferences (Liu et al. 2025). This creates a significant barrier to adaptation, particularly for larger models with billions of parameters, where retraining for each preference configuration becomes prohibitively expensive (Wu et al. 2025; Zhang et al. 2026b,a).

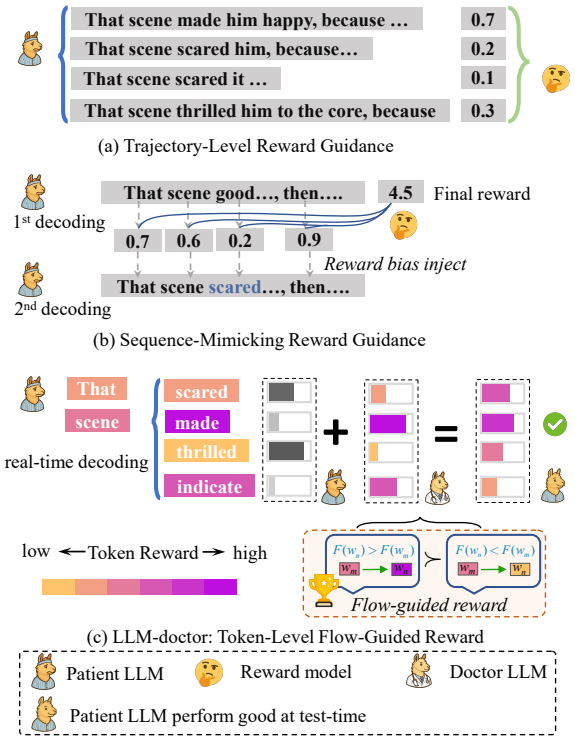

Figure 1: Comparison of test-time alignment approaches.

Test-time alignment methods (Shen et al. 2025b,c; Hua et al. 2025) address these limitations by guiding frozen LLMs during inference without modifying their underlying weights. Within this paradigm, reward-guided approaches have emerged as a promising direction, where a smaller reward model (RM) steers the generation of a larger frozen LLM (Zhou et al. 2024; Shen et al. 2025a). As shown in Fig.1, these approaches aim to maintain the LLM's generative capabilities while enabling flexible alignment with specific objectives through adjustable guidance signals at inference time, potentially accommodating different alignment goals without repeated training (Lin et al. 2025).

*Corresponding author

Conventional reward-guided test-time alignment methods face fundamental limitations in their preference modeling. Trajectory-level evaluation methods, as shown in Fig.1 (a), rely on trajectory-level reward models that evaluate complete sequences or trajectories (Ouyang et al. 2022; Yuan et al. 2025). This approach inevitably necessitates multiple sampling iterations to generate diverse candidate responses, resulting in substantial computational overhead from producing numerous invalid or low-quality text sequences. To address these inefficiencies, sequence-mimicking methods in Fig.1 (b) train reward models to assign token-level scores that aim to reflect trajectory-level preferences. However, the sequence-mimicking reward guidance approach is fundamentally limited by its training objective. Since the method relies on a single preference score for an entire trajectory, the reward model must distribute this score across all constituent tokens, often to satisfy a "reward-budget" constraint (Xu et al. 2025). This mechanical distribution creates unreliable and non-local credit assignment, for instance, the model may assign artificially high rewards to neutral tokens (e.g., connectives like "and" or "the") simply to ensure the total score for a preferred sequence is higher, it dilutes the optimization signal from the few tokens that are actually critical to human preference, thereby hindering optimization (Shao et al. 2025; Pang et al. 2025). This distortion is compounded by a theoretical ceiling effect: the larger model being guided converges to mimicking the smaller reward model, thus capping performance at the reward model's limited capabilities and negating the superior capabilities of the larger base LLM.

This motivates the exploration of a new alignment paradigm: one that can directly assess the preference contribution of individual tokens, thereby preserving the base model's inherent capabilities while avoiding the limitations of trajectory-level reward allocation. To this end, this paper introduces LLMdoctor, a three-stage framework that integrates token-level rewards with flow-guided optimization for efficient and effective test-time alignment. As shown in Fig.2, the framework begins with token-level reward acquisition, where we extract token-level reward signals by analyzing behavioral variations of the *patient* model (the large frozen LLM) on human preference data. Unlike conventional approaches that treat entire sequences as atomic units, LLMdoctor identifies specific tokens that significantly contribute to preference judgments, thereby producing a fine-grained and reliable reward signal. Given that each token reward is computed from the *context-dependent log-likelihood gap* between a POSITIVE and a NEGATIVE behavioural variant of the same *patient* model, our scheme assigns rewards only to genuinely discriminative tokens instead of forcing all per-token scores to balance to a preset trajectory total. This contrastive, sparsity-controlled signal sidesteps the compensatory "reward-budget" distortion suffered by sequence-mimicking methods and lays a faithful foundation for the subsequent flow-guided optimization stage. These token-level rewards then serve as training signals for token-level flow-guided preference optimization (TFPO). TFPO enforces flow conservation across all subtrajectories. This approach expands the preference signal from $\mathcal{O}(1)$ at the trajectory level to $\mathcal{O}(n^2)$ at the subtrajectory level, creating a comprehensive token-by-token alignment mechanism. Its flow balance constraints naturally maintain diversity in generation trajectories, preventing the mode collapse common in reward-maximizing approaches and preserving the rich generative capabilities of the original model. Finally, the *doctor* model guides the *patient* model at inference time as a flow-guided reward model, providing token-level preference signals that inform the *patient* model's generation process.

The contributions of this work are three-fold: (1) We introduce a test-time alignment framework that extracts and leverages fine-grained token-level rewards, providing direct preference signals without relying on trajectory-level reward models. (2) We propose token-level TFPO, a method that expands preference signals to the subtrajectory level to train a novel flow-guided reward model. (3) Our approach supports multi-dimensional preference alignment, enabling real-time adjustment of different alignment objectives without retraining. Experiments on multiple domains demonstrate that LLMdoctor significantly outperforms existing test-time alignment methods while matching or exceeding the performance of more costlier training-time approaches.

2 Preliminaries

Generative Flow Networks (GFlowNets) (Bengio et al. 2023) introduce the principle of flow balance for learning to sample complex discrete objects: Each partially constructed object (a state) must maintain an equilibrium between incoming and outgoing *flow*, which can be conceptualized as a measure of trajectory density through that state. For any non-terminal state s in the generation process, the total flow entering s from its predecessor states must equal the total flow exiting s towards its successor states:

$$\sum_{s' \in \text{Pred}(s)} F(s' \to s) = \sum_{s'' \in \text{Succ}(s)} F(s \to s''), \qquad (1)$$

where $F(s_a \to s_b)$ denotes the flow associated with the transition from state s_a to state s_b. Furthermore, the flow terminating at a complete object (terminal state s_L) is typically set to be proportional to a reward or energy function $R(s_L)$ associated with that object: $F(s_L) \propto R(s_L)$.

Traditional preference optimization methods for LLMs, such as RLHF and DPO, often evaluate preferences at the entire response level. This can overlook the nuanced contributions of individual tokens to the overall quality of a generated sequence.The LLMdoctor framework, particularly through its token-level TFPO stage (Section 3.2), adapts the flow balance concept to the autoregressive token generation process. By associating flow with token sequence prefixes, TFPO aims to ensure that the generation of each token aligns with preference signals. The probability of generating a sequence of tokens that extends a prefix s_m to a longer prefix s_n is determined by the ratio of their respective flows:

$$P(s_m \rightsquigarrow s_n) \;\propto\; \frac{F(s_n)}{F(s_m)}, \qquad (2)$$

where $s_m \rightsquigarrow s_n$ denotes the generation of the token subsequence from s_m to s_n. This flow-guided mechanism encourages a model to allocate higher probability mass to continuations with greater downstream flow, thereby promoting

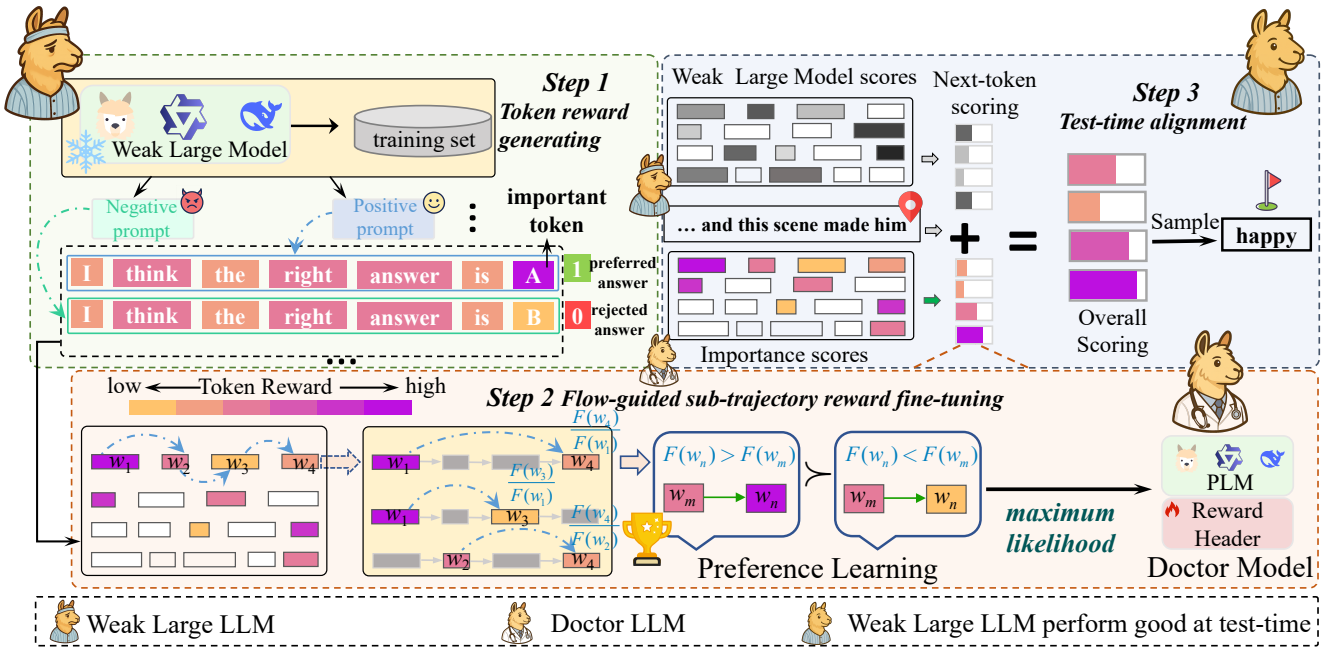

Figure 2: Overall framework of LLMdoctor

preference-aligned generation at each step of the autoregressive process.

3 Methodology

We introduce a novel framework for LLM alignment using token-level rewards at inference time. This approach addresses three critical challenges in current alignment methods: 1) obtaining fine-grained token-level supervision signals, 2) reducing computational overhead in preference optimization, and 3) enabling flexible alignment during generation. Fig. 2 illustrates our proposed architecture. The framework operates through a three-stage process linking a large pre-trained *patient* model with a smaller *doctor* model.

First, the **token-level reward generating** stage extracts detailed reward signals by analyzing the *patient* model's responses to various prompts informed by human preference data. These token-level rewards then serve as training signals for **flow-guided sub-trajectory reward fine-tuning** of the *doctor* model. This stage employs flow-guided direct preference optimization to establish token-by-token preference alignment (TFPO) within the smaller model. Finally, during **test-time alignment** at online alignment stage, the trained small *doctor* model dynamically guides the *patient* model's outputs at inference time, eliminating the need to retrain the larger model. This integration creates an efficient alignment pipeline by concentrating intensive training on the smaller *doctor* model while preserving the generative capabilities of the *patient* model. The approach enables flexible preference adjustment during inference without expensive retraining, creating a practical solution for aligning large-scale language models with human preferences at test time.

3.1 Token-Level Reward Acquisition

The token-level reward acquisition stage begins with an LLM that has undergone supervised fine-tuning but not preference alignment, serving as the *patient* model. This stage extracts fine-grained token-level signals by analyzing the model's behavioral responses to prompts from a standard preference dataset, $\mathcal{D} = \{(x^{(i)}, y_+^{(i)}, y_-^{(i)})\}_{i=1}^N$, where each instance contains a prompt $x^{(i)}$, a human-preferred response $y_+^{(i)}$, and a non-preferred response $y_-^{(i)}$. Instead of training separate reward models, LLMdoctor creates behavioral variants of the *patient* model via conditioning, revealing token importance by measuring differences in log-probabilities assigned to tokens under contrasting behaviors.

The importance measurement is then combined with human preference labels to determine the magnitude and direction of token-level rewards, reinforcing important tokens in preferred responses while suppressing them in non-preferred ones.

Behavioral Variants from a Single Model. The *patient* model π_{SFT} serves as the foundation for creating discriminative behavioral variants. Through strategic prompt engineering, the model generates two distinct behavioral modes without requiring additional parameters or training, namely a positive face π^{pos} (a variant instructed to generate helpful, accurate, and polite responses), and a negative face π^{neg} (a variant prompted to produce less helpful responses with critical information omitted). These variants share the same parameters but exhibit different response distributions based on their prompting.

Token Importance Measurement. For each token y_t at position t in a response y (which can be either a preferred response $y_+^{(i)}$ or a non-preferred response $y_-^{(i)}$ from an in-

stance $(x^{(i)}, y_+^{(i)}, y_-^{(i)})$ in the **training split** of the preference dataset $\mathcal{D}$), the importance estimation process computes log-likelihoods under both behavioral variants:

$$\ell_t^{\text{pos}} = \log \pi^{\text{pos}}(y_t \mid x, y_{<t}), \quad \ell_t^{\text{neg}} = \log \pi^{\text{neg}}(y_t \mid x, y_{<t}). \tag{3}$$

The absolute difference $\Delta_t = |\ell_t^{\text{pos}} - \ell_t^{\text{neg}}|$ measures how strongly each token distinguishes between positive and negative behaviors. Tokens with larger differences play more significant roles in determining response quality. This direct measure of behavioral distinctiveness thus avoids misattributing high importance to tokens that are frequent but not genuinely discriminative. To ensure comparability across different response styles and lengths, the raw differences undergo normalization and smoothing:

$$\widehat{\Delta}_t = \frac{\Delta_t}{\text{mean}_j(\Delta_j) + \varepsilon}, \quad S_t = \tanh\Big(\frac{\widehat{\Delta}_t}{\tau}\Big), \tag{4}$$

where ε is a small constant that prevents division by zero, and τ is a temperature parameter controlling the smoothness of importance scores. The final score $S_t \in (0, 1)$ represents each token's importance in distinguishing between desired and undesired behaviors.

Token-Level Reward Assignment. Directional token rewards are obtained by combining importance scores with binary human preference signals $\text{sign}(y) \in \{+1, -1\}$:

$$r_t = \text{sign}(y) \cdot S_t \cdot \mathbf{1}[S_t > \theta], \tag{5}$$

where $\mathbf{1}[\cdot]$ is an indicator function and θ is a sparsity threshold. This formulation ensures that only substantially discriminative tokens receive non-zero rewards, with the magnitude reflecting importance and the sign indicating whether to reinforce or suppress the token. These token-level rewards provide a fine-grained supervision signal for the subsequent training of the *doctor* model. By operating at the token level, the framework identifies the specific tokens that contribute most to human preferences, enabling precise and localized credit assignment.

3.2 TFPO-Based Fine-Grained Preference Tuning

Given token-level rewards r_t from the *patient* model, the smaller *doctor* model $\hat{\pi}_\theta$ is now trained to internalize these fine-grained alignment signals via token-level TFPO. Token-level TFPO extends preference optimization to the subtrajectory level within token sequences. It incorporates a value function V_ϕ, which is a head of the *doctor* model, to estimate the value of token sequence prefixes.

Flow-Guided Optimization for Token Sequences. The TFPO framework views token generation as a trajectory through states. A state s_t represents the sequence of t tokens $(y_1, \ldots, y_t)$ generated thus far, with s_0 denoting the initial prompt context. The *doctor* model $\hat{\pi}_\theta(y_{t+1}|s_t)$ defines the probability of generating the next token y_{t+1} given the current state (prefix) s_t. TFPO builds on the flow conservation principle from GFlowNets. The *flow* $F(s_t)$ through a state s_t represents the unnormalized probability mass passing through that prefix. This flow is defined as the product of a prefix score $Q(s_t)$, derived from token-level rewards, and a learned value estimate $V_\phi(s_t)$ that discriminates among candidate continuations:

$$F(s_t) = Q(s_t) \cdot V_\phi(s_t), \tag{6}$$

where $Q(s_t)$ is a positive weighting term derived from the token-level rewards r_k (for $k < t$) obtained from the *patient* model, encoding the preference information associated with the prefix s_t.

The flow conservation principle dictates that for any non-terminal state s_t, the total incoming flow must equal the total outgoing flow. The probability of transitioning from a prefix s_m to a longer prefix s_n (by appending tokens $y_m, \ldots, y_{n-1}$) equals the ratio of their flows, $F(s_n)/F(s_m)$, representing the share of the parent's flow allocated to this continuation. This naturally creates a *flow allocation* effect: among multiple candidate continuations from the same prefix, those with higher downstream flow receive larger probability shares, thereby directing the policy $\hat{\pi}_\theta$ toward more preferred branches.

Subtrajectory Balance Objective for TFPO. This flow balance requirement is formalized through the Subtrajectory Balance (SubTB) principle. For any generation trajectory $\tau : s_0 \xrightarrow{y_1} s_1 \ldots \xrightarrow{y_L} s_L$ (where s_0 is the initial prompt context and L is the sequence length), and for any subtrajectory from state s_m to s_n (where $0 \le m < n \le L$), the SubTB condition, assuming a forward policy $\hat{\pi}_\theta$ (the *doctor* model) and a backward policy $\hat{\pi}_B$, is given by:

$$F(s_m) \prod_{k=m}^{n-1} \hat{\pi}_\theta(y_{k+1}|s_k) = F(s_n) \prod_{k=m}^{n-1} \hat{\pi}_B(y_k|s_{k+1}). \tag{7}$$

This equation ensures that the **forward flow** from s_m to s_n matches the **backward flow**.

Following common practice in GFlowNet formulations for sequence generation, a uniform backward policy ($\hat{\pi}_B(\cdot) = 1$) is adopted without loss of generality, as the primary goal is to learn the forward generative policy $\hat{\pi}_\theta$. Substituting Eq. 6 into Eq. 7 and setting $\hat{\pi}_B = 1$ yields:

$$Q(s_m)V_\phi(s_m) \prod_{k=m}^{n-1} \hat{\pi}_\theta(y_{k+1}|s_k) = Q(s_n)V_\phi(s_n). \tag{8}$$

This condition implies that the cumulative probability of generating the token sequence from s_m to s_n equals the flow ratio $F(s_n)/F(s_m)$, which represents the fraction of the source state's flow allocated to this specific continuation. Consequently, among different candidate continuations from the same prefix s_m, those leading to states with higher composite flow will receive proportionally larger probability mass.

To derive a trainable loss function, we take the logarithm of both sides of Eq. 8 and rearrange terms, leading to:

$$\log \frac{Q(s_n)V_\phi(s_n)}{Q(s_m)V_\phi(s_m)} = \sum_{k=m}^{n-1} \log \hat{\pi}_\theta(y_{k+1}|s_k). \tag{9}$$

The Subtrajectory Balance loss for TFPO ($\mathcal{L}_{\text{SubTB}}$) penalizes the squared difference from this equality over all possible

subtrajectories within each sequence in the training dataset $\mathcal{D}_{pref}$ (derived from the original preference data $\mathcal{D}$):

$$\mathcal{L}_{\text{SubTB}}(\hat{\pi}_\theta, V_\phi) = \sum_{(\tau)\in\mathcal{D}_{pref}} \sum_{0\le m<n\le L_\tau} \left(\log \frac{Q(s_n)V_\phi(s_n)}{Q(s_m)V_\phi(s_m)} - \sum_{k=m}^{n-1} \log \hat{\pi}_\theta(y_{k+1}|s_k) \right)^2, \tag{10}$$

where L_τ is the length of trajectory τ. This loss trains the *doctor* model $\hat{\pi}_\theta$ and the value function V_ϕ to satisfy flow consistency across all token subsequences, guided by the prefix scores $Q(s_t)$ derived from the *patient* model's token-level rewards.

Value Discrimination Loss. To further ensure that the value function V_ϕ correctly distinguishes between more and less preferred next tokens based on the initial token-level rewards, a value discrimination loss is employed. Given a prefix s_t, if token y_w is considered preferable to y_l (e.g., $r(y_w) > r(y_l)$ from *patient* model feedback), the value loss encourages V_ϕ to reflect:

$$\mathcal{L}_{\text{value}}(V_\phi) = \max(0, \gamma - (V_\phi(s_t, y_w) - V_\phi(s_t, y_l))), \tag{11}$$

where (s_t, y_w) denotes the state (prefix) resulting from appending y_w to s_t, and γ is a margin hyperparameter. This requires V_ϕ to estimate the value of a prefix after a specific next token is chosen.

Overall TFPO Training Objective. The training objective for the *doctor* model using TFPO combines the subtrajectory balance loss and the value discrimination loss:

$$\mathcal{L}_{\text{TFPO}} = \mathcal{L}_{\text{SubTB}}(\hat{\pi}_\theta, V_\phi) + \lambda \mathcal{L}_{\text{value}}(V_\phi), \tag{12}$$

where λ is a hyperparameter that balances the contribution of the two loss components.

Training Procedure. The training of the *doctor* model $\hat{\pi}_\theta$ and its value head V_ϕ commences after acquiring the token-level rewards r_t (which inform prefix scores $Q(s_t)$) from the *patient* model's analysis of the preference dataset $\mathcal{D}_{pref}$, as detailed in Section 3.1. Using these pre-computed rewards, the *doctor* model parameters are then optimized by minimizing the overall TFPO objective $\mathcal{L}_{\text{TFPO}}$ (Eq. 12).

This procedure enables the *doctor* model to learn token-level preference alignment by satisfying flow balance conditions across entire generation trajectories, thereby developing a context-aware ability to dynamically evaluate the preference alignment of potential next tokens while preserving generation diversity.

3.3 Online Alignment

The LLMdoctor framework ends with the Online Alignment stage, where the trained *doctor* model guides the *patient* model's output during inference.

Flow-Guided Reward Model Formulation. The trained *doctor* model is employed as a flow-guided reward model. Given a generation context and the sequence of tokens produced so far (state $s_t = (y_1, \ldots, y_t)$), the flow-guided reward model outputs a log-probability score, $\log \pi_r(y_{t+1}|s_t)$, for each potential next token y_{t+1}. These scores function as dynamic, token-level preference signals that inform the *patient* model's generation process.

Reward-Guided Decoding Algorithm. At inference, the *patient* model's log-probabilities (π_{base}) are combined with the token-level preference signals from the flow-guided reward model (π_r) to derive a modified decoding distribution:

$$\pi_{\text{decode}}(y_{t+1} \mid s_t) \propto \left[\pi_{\text{base}}(y_{t+1} \mid s_t)\right]^\alpha \cdot \left[\pi_r(y_{t+1} \mid s_t)\right]^\beta, \tag{13}$$

where α and β are adjustable hyperparameters that control the trade-off between fluency and preference alignment.

This mechanism is computationally efficient, as both models compute their respective distributions for all candidate next tokens in a single forward pass. This obviates the need for multiple full-sequence generations for evaluation.

Flexible Online Alignment. Our framework can be used for multi-dimensional preference control, e.g., balancing helpfulness and safety. To achieve this, we can train specialized *doctor* models for each preference dimension (or develop a unified model with separate reward heads for each aspect). During inference, guidance from these models is integrated by modifying the decoding process:

$$\pi_{\text{decode}}(y_{t+1} \mid s_t) \propto \left[\pi_{\text{base}}(y_{t+1} \mid s_t)\right]^\alpha \cdot \prod_i \left[\pi_r^{(i)}(y_{t+1} \mid s_t)\right]^{\beta_i} \tag{14}$$

where $\pi_r^{(i)}$ represents the flow-guided reward model for the i-th dimension, and β_i are adjustable weights. This configuration permits dynamic balancing of different alignment aspects at inference time by modifying the β_i coefficients, without the need to retrain either the large *patient* model or the specialized *doctor* models.

4 Experiments

4.1 Experimental Setup

Datasets. HH-RLHF (Helpful and Harmless) (Bai et al. 2022): comprising 112,000 training samples and 12,500 test samples for general alignment evaluation. PKU-SafeRLHF-10K (Ji et al. 2024): including explicit preference labels for both helpfulness and harmlessness dimensions separately. UltraFeedback (Cui et al. 2023): providing extensive preference data for training reward models.

Baselines. The performance of LLMdoctor is benchmarked against a comprehensive suite of established methods spanning multiple categories. **1) For standard decoding**, we use greedy search, top-k sampling, top-p (nucleus) sampling, and contrastive search. **2) For training-time alignment**, we compare with Direct Preference Optimization (DPO) (Rafailov et al. 2023). **3) For test-time alignment**, we evaluate against methods including Autoregressive Reward Search (ARGS) (Khanov, Burapacheep, and Li 2024), Generative Autoregressive Reward Modeling (GenARM) (Xu et al. 2025), and Naive Rejection Sampling (Naive RS) (Li et al. 2024). **4) For multi-objective alignment**, we compare against approaches such as Reward Soups (RS) (Rame et al. 2023) and Multi-objective RL (MORL) (Wu et al. 2023).

Models and Training. For most experiments, we follow the settings of ARGS (Khanov, Burapacheep, and Li 2024) and

Method vs. Method	Win (%)	Tie (%)	Lose (%)	Win + ½ Tie (%)†
ARGS vs. DPO	24.54±0.17	3.39±0.32	72.07±0.30	26.24±0.17
Transfer-Q vs. DPO	31.30±0.30	4.14±0.17	64.56±0.18	33.37±0.22
CARDS vs. DPO	38.29±0.17	6.51±0.16	55.20±0.31	41.55±0.23
GenARM vs. DPO	49.60±0.31	5.29±0.17	45.11±0.34	52.25±0.32
GenARM vs. ARGS	67.53±0.51	6.02±0.33	26.45±0.17	70.54±0.35
GenARM vs. Transfer-Q	67.82±0.35	4.39±0.17	27.79±0.18	70.02±0.26
GenARM vs. CARDS	56.47±0.14	3.82±0.32	39.71±0.35	58.38±0.17
Ours vs. Greedy Search	89.40±0.25	7.10±0.18	3.50±0.15	92.95±0.21
Ours vs. Top-k Sampl.	87.20±0.28	8.30±0.21	4.50±0.16	91.35±0.24
Ours vs. Top-p Sampl.	86.80±0.29	8.90±0.22	4.30±0.15	91.25±0.25
Ours vs. Contra. Search	81.50±0.35	10.20±0.25	8.30±0.22	86.60±0.31
Ours vs. Naive RS	76.60±0.41	11.40±0.28	12.00±0.29	82.30±0.37
Ours vs. DPO	57.80±0.33	6.40±0.19	35.80±0.31	61.00±0.30
Ours vs. ARGS	73.20±0.45	5.60±0.18	21.20±0.38	76.00±0.39
Ours vs. Transfer-Q	74.10±0.42	4.50±0.16	21.40±0.37	76.35±0.36
Ours vs. CARDS	69.50±0.48	5.90±0.20	24.60±0.41	72.45±0.42
Ours vs. GenARM	58.50±0.35	7.20±0.21	34.30±0.32	62.10±0.32

Table 1: Head-to-head comparison on the HH-RLHF test set, evaluated by GPT-4o. Cell color intensity indicates win/loss magnitude (purple for win, orange for loss). †Win + ½ Tie percentages are reported as a summary statistic.

use the LLaMA-7B-SFT checkpoint as the base LLM, fine-tuning it with LoRA on the HH-RLHF training split to create reward models for test-time methods. For the weak-to-strong guidance experiments, we use the Tulu2 model family (Ivison et al. 2023), specifically the supervised fine-tuned (SFT) checkpoints at 7B, 13B, and 70B parameter scales. For LLMdoctor, the *doctor* model is trained as described in Section 3.2. DPO is trained by fine-tuning the corresponding SFT model on the relevant preference dataset. Parameters for baseline methods are set according to their original papers or tuned on a validation set for fair comparison.

Evaluation. Following the protocol of Khanov, Burapacheep, and Li (2024) and Xu et al. (2025), responses are generated for 300 randomly sampled prompts from the HH-RLHF test set, with alignment performance evaluated using head-to-head comparisons judged by GPT-4o. For the weak-to-strong guidance experiments, we use AlpacaEval 2 (Dubois et al. 2024), an automatic evaluation framework that compares model outputs against a reference model and computes win rates.

4.2 Main Results

We evaluate alignment performance using head-to-head comparisons judged by GPT-4o, with the "Win + ½ Tie (%)" metric serving as the primary measure, summarized in Table 1. LLMdoctor demonstrates a consistent and significant advantage over all baselines. Critically, its superiority extends across alignment paradigms, surpassing the strongest test-time method, GenARM, and outperforming the full training-time approach, DPO. Notably, other test-time methods like ARGS (26.24%), Transfer-Q (33.37%), and CARDS (41.55%) exhibit a significant performance gap against DPO. Furthermore, LLMdoctor overwhelmingly outperforms standard unaligned decoding strategies, such as Naive RS (82.30%) and top-p sampling (91.25%).This consistent outperformance validates LLMdoctor's token-level flow-guided optimization.

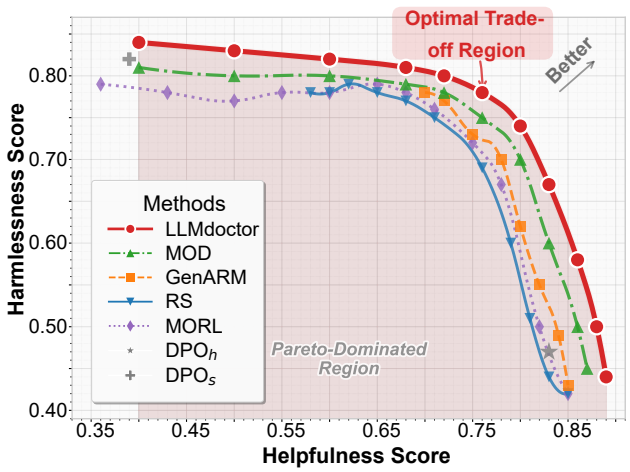

Figure 3: Pareto frontier comparison for helpfulness and harmlessness.

4.3 Multi-Dimensional Preference Balancing

Real-world preference alignment often requires navigating multiple, potentially conflicting dimensions. To evaluate LLMdoctor's capability in balancing helpfulness and harmlessness, we conduct a Pareto frontier analysis on the PKU-SafeRLHF-10K dataset. For this task, we train specialized *doctor* models for the helpfulness and harmlessness dimensions respectively. During inference, their guidance is dynamically combined using adjustable weights (β_h, β_s), allowing us to trace a Pareto frontier by systematically varying their balance.

As shown in Fig. 3, LLMdoctor's frontier consistently dominates other methods, achieving superior trade-offs across all parameter configurations. Unlike training-based methods that require retraining for different preference configurations, LLMdoctor enables real-time adjustment of preference weights during inference, highlighting its flexibility.

4.4 Weak-to-Strong Guidance

To evaluate LLMdoctor's efficacy in a weak-to-strong guidance scenario, a 7B *doctor* model guides *patient* models of increasing scale (Tulu2-SFT at 7B, 13B, and 70B). The performance is benchmarked against other test-time methods, which also employ a 7B guidance model, and against DPO, which requires full fine-tuning at each respective scale. To ensure a controlled comparison, all methods are evaluated by their win rates against a fixed Tulu2-7B SFT reference model using the AlpacaEval 2 benchmark.

As shown in Fig. 4, LLMdoctor consistently outperforms other test-time alignment methods across all *patient* model scales. Notably, the 7B *doctor* model surpasses the fully fine-tuned DPO baselines at every scale, achieving a length-controlled win rate of 82.5% at the 70B scale compared to DPO's 82.0%. This demonstrates that the proposed framework can effectively transfer alignment capabilities from smaller to larger models without incurring the substantial computational cost of fine-tuning.

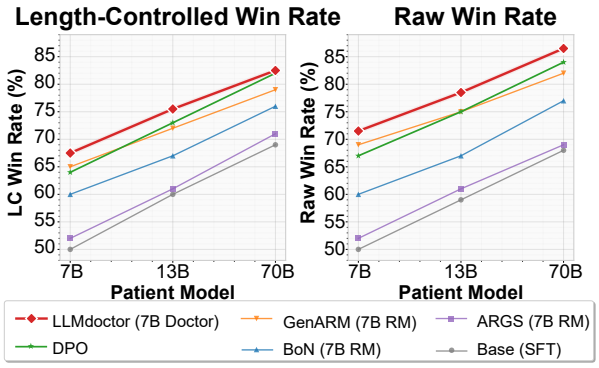

Figure 4: Weak-to-strong guidance performance. Comparison of length-controlled (LC) and raw AlpacaEval 2 win rates across different base model scales. All test-time methods employ a 7B guidance model, while DPO involves full fine-tuning at each respective scale.

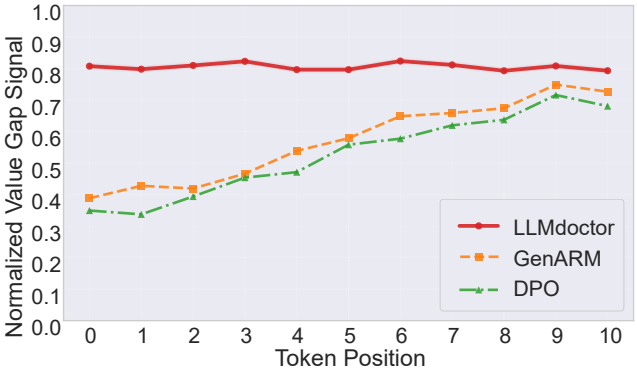

Figure 5: Alignment signal dynamics.

4.5 Alignment Signal Dynamics Analysis

To investigate how different alignment methods guide generation over time, we analyze their internal alignment signals. At each step of generating a preferred response, we measure a "value gap" that quantifies how confidently a model distinguishes the correct next token from a plausible alternative predicted by the base SFT model. A larger gap signifies a stronger, more decisive alignment signal, indicating better foresight. Fig. 5 highlights distinct patterns in the signal dynamics. LLMdoctor maintains a consistently high normalized signal throughout the generation process. This suggests that the TFPO mechanism successfully propagates sequence-level preference information to each intermediate step, providing the *doctor* model with strong "foresight" from the beginning. In contrast, DPO and GenARM both exhibit "climbing" trajectories, where signals start at a lower level and gradually strengthen as more tokens are generated.

4.6 Performance vs. Diversity Analysis

This section analyzes the trade-off between alignment performance and generation diversity for the 7B models on the HH-RLHF dataset. Performance is measured by win rates against DPO.

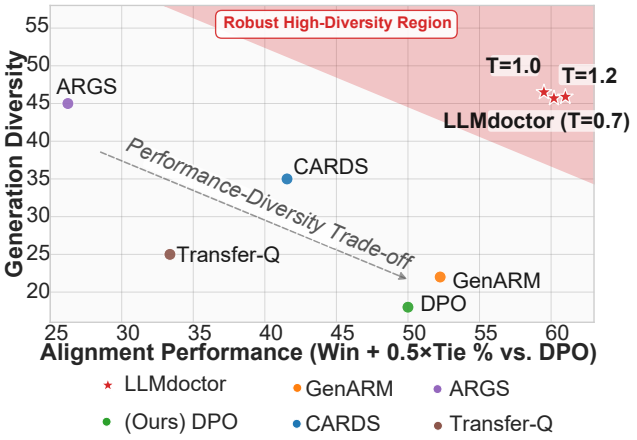

Figure 6: Performance vs. diversity trade-off. The plot compares alignment performance (Win + 0.5×Tie % vs. DPO) against generation diversity for various methods.

Method Variant	Win + ½ Tie (%) vs. DPO	Diversity
LLMdoctor (Full Model)	**61.00**	**0.47**
w/o Subtrajectory Balance ($\mathcal{L}_{\text{SubTB}}$)	53.15	0.34
w/o Value Discrimination ($\mathcal{L}_{\text{value}}$)	58.23	0.43
w/o Reward Sparsity	56.58	0.46
w/o Flow-Guided Rewards	52.76	0.25

Table 2: Ablation study results on the HH-RLHF test set.

The results in Fig. 6 reveal that LLMdoctor excels in both dimensions, achieving the highest alignment score while maintaining superior diversity over other test-time methods. In contrast, ARGS preserves high diversity at the cost of performance, while GenARM and Transfer-Q sacrifice diversity for alignment gains. DPO exhibits the lowest diversity, consistent with the known mode collapse tendency of training-time methods. This analysis empirically confirms that LLMdoctor's flow-guided optimization effectively achieves strong alignment without compromising the base model's generative richness.

4.7 Ablation Study

As shown in Table 2, the ablation experiments demonstrate the effectiveness of the method proposed in this paper.

5 Conclusion

This paper introduces LLMdoctor, a novel framework to enhance test-time alignment of large language models. LLMdoctor employs a patient-doctor paradigm where a smaller doctor model, trained with token-level flow-guided preference optimization (TFPO), provides real-time guidance to a large, frozen patient model. This approach enables flexible and efficient alignment without costly retraining. Experiments demonstrate that LLMdoctor significantly outperforms existing alignment methods in both preference alignment and generation diversity, highlighting the potential of flow-based optimization to create more powerful, adaptable alignment solutions for state-of-the-art language models.

Acknowledgments

This research is supported by the RIE2025 Industry Alignment Fund – Industry Collaboration Projects (IAF-ICP) (Award I2301E0026), administered by A*STAR, as well as supported by Alibaba Group and NTU Singapore through Alibaba-NTU Global e-Sustainability CorpLab (ANGEL). The work is also supported by the Ministry of Education, Singapore under its MOE Academic Research Fund Tier 2 (MOE-T2EP20123-0005).

References

Bai, Y.; Jones, A.; Ndousse, K.; Askell, A.; Chen, A.; DasSarma, N.; Drain, D.; Fort, S.; Ganguli, D.; Henighan, T.; et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Bengio, Y.; Lahlou, S.; Deleu, T.; Hu, E. J.; Tiwari, M.; and Bengio, E. 2023. GFlowNet Foundations. *Journal of Machine Learning Research*, 24(210): 1–55.

Cui, G.; Yuan, L.; Ding, N.; Yao, G.; He, B.; Zhu, W.; Ni, Y.; Xie, G.; Xie, R.; Lin, Y.; et al. 2023. ULTRAFEEDBACK: Boosting Language Models with Scaled AI Feedback. In *Forty-first International Conference on Machine Learning*.

Dubois, Y.; Galambosi, B.; Liang, P.; and Hashimoto, T. B. 2024. Length-Controlled AlpacaEval: A Simple Way to Debias Automatic Evaluators. arXiv:2404.04475.

Hua, Y.; Qu, L.; Li, Z.; Xue, H.; Salim, F. D.; and Haffari, G. 2025. RIDE: Enhancing Large Language Model Alignment through Restyled In-Context Learning Demonstration Exemplars. *arXiv preprint arXiv:2502.11681*.

Ivison, H.; Wang, Y.; Pyatkin, V.; Lambert, N.; Peters, M.; Dasigi, P.; Jang, J.; Wadden, D.; Smith, N. A.; Beltagy, I.; and Hajishirzi, H. 2023. Camels in a Changing Climate: Enhancing LM Adaptation with Tulu 2. arXiv:2311.10702.

Ji, J.; Hong, D.; Zhang, B.; Chen, B.; Dai, J.; Zheng, B.; Qiu, T.; Li, B.; and Yang, Y. 2024. Pku-saferlhf: A safety alignment preference dataset for llama family models. *arXiv e-prints*, arXiv–2406.

Khanov, M.; Burapacheep, J.; and Li, Y. 2024. Args: Alignment as reward-guided search. *arXiv preprint arXiv:2402.01694*.

Li, B.; Wang, Y.; Grama, A.; and Zhang, R. 2024. Cascade Reward Sampling for Efficient Decoding-Time Alignment. In *ICML 2024 Next Generation of AI Safety Workshop*.

Lin, B.; Jiang, W.; Xu, Y.; Chen, H.; and Chen, Y.-C. 2025. PARM: Multi-Objective Test-Time Alignment via Preference-Aware Autoregressive Reward Model. arXiv:2505.06274.

Liu, S.; Fang, W.; Hu, Z.; Zhang, J.; Zhou, Y.; Zhang, K.; Tu, R.; Lin, T.-E.; Huang, F.; Song, M.; Li, Y.; and Tao, D. 2025. A Survey of Direct Preference Optimization. arXiv:2503.11701.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C. L.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; Schulman, J.; Hilton, J.; Kelton, F.; Miller, L.; Simens, M.; Askell, A.; Welinder, P.; Christiano, P.; Leike, J.; and Lowe, R. 2022. Training language models to follow instructions with human feedback. arXiv:2203.02155.

Pang, J.; Di, N.; Zhu, Z.; Wei, J.; Cheng, H.; Qian, C.; and Liu, Y. 2025. Token Cleaning: Fine-Grained Data Selection for LLM Supervised Fine-Tuning. arXiv:2502.01968.

Rafailov, R.; Sharma, A.; Mitchell, E.; Ermon, S.; Manning, C. D.; and Finn, C. 2023. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. arXiv:2305.18290.

Rame, A.; Couairon, G.; Dancette, C.; Gaya, J.-B.; Shukor, M.; Soulier, L.; and Cord, M. 2023. Rewarded soups: towards pareto-optimal alignment by interpolating weights fine-tuned on diverse rewards. *Advances in Neural Information Processing Systems*, 36: 71095–71134.

Shao, R.; Li, B.; Liu, G.; Chen, Y.; Zhou, X.; Wang, J.; Cai, X.; and Li, P. 2025. EARLIER TOKENS CONTRIBUTE MORE: LEARNING DIRECT PREFERENCE OPTIMIZATION FROM TEMPORAL DECAY PERSPECTIVE. *Published as a conference paper at ICLR 2025*.

Shen, T.; Mao, R.; Wang, J.; Zhang, X.; and Cambria, E. 2025a. Flow-guided Direct Preference Optimization for Knowledge Graph Reasoning with Trees. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1165–1175.

Shen, T.; Wang, J.; Zhang, X.; and Cambria, E. 2025b. Hoplevel Direct Preference Optimization for Knowledge Graph Reasoning with Trees. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5. IEEE.

Shen, T.; Wang, J.; Zhang, X.; and Cambria, E. 2025c. Reasoning with trees: faithful question answering over knowledge graph. In *Proceedings of the 31st International Conference on Computational Linguistics*, 3138–3157.

Wu, J.; Huang, K.; Wang, X.; Gao, J.; Ding, B.; Wu, J.; He, X.; and Wang, X. 2025. RePO: ReLU-based Preference Optimization. arXiv:2503.07426.

Wu, Z.; Hu, Y.; Shi, W.; Dziri, N.; Suhr, A.; Ammanabrolu, P.; Smith, N. A.; Ostendorf, M.; and Hajishirzi, H. 2023. Fine-grained human feedback gives better rewards for language model training. In *Advances in Neural Information Processing Systems*, volume 36.

Xu, Y.; Sehwag, U. M.; Koppel, A.; Zhu, S.; An, B.; Huang, F.; and Ganesh, S. 2025. GenARM: Reward Guided Generation with Autoregressive Reward Model for Test-time Alignment. arXiv:2410.08193.

Yuan, L.; Cai, Y.; Shen, X.; Li, Q.; Huang, Q.; Deng, Z.; and Wang, T. 2025. Collaborative Multi-LoRA Experts with Achievement-based Multi-Tasks Loss for Unified Multimodal Information Extraction. In Kwok, J., ed., *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25*, 6940–6948. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Zhang, J.; Wang, Z.; Wang, Z.; Zhang, X.; Xu, F.; Lin, Q.; Mao, R.; Cambria, E.; and Liu, J. 2026a. MAPS: A multiagent framework based on big seven personality and socratic

guidance for multimodal scientific problem solving. In *Proceedings of AAAI*.

Zhang, J.; Wang, Z.; Zhu, H.; Liu, J.; Lin, Q.; and Cambria, E. 2026b. MARS: A multi-agent framework incorporating socratic guidance for automated prompt optimization. In *Proceedings of AAAI*.

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. arXiv:2405.19262.