Adaptive Prompt Optimization by Self-Evolution: a Task Entropy Preference Approach

Anonymous ACL submission

Abstract

We study what kinds of prompt contexts improve LLM performance in specific cases and what fails. Our hypothesis is that a context helps only when the distribution shift it induces at decoding aligns with the task's entropy preference: reliability-oriented tasks (e.g., trustworthy alignment and jailbreak resistance) favor lower generative entropy, whereas creative/open-ended tasks favor higher but controlled entropy. We evaluate this hypothesis across safety, instruction following, and creativity scenarios, analyzing the relationship between token-level semantic entropy and task metrics under multiple decoding policies. Guided by these observations, we introduce a training-free, self-evolving prompt optimizer that probes a task with paired low- vs. high-entropy contexts, uses an LLM judge with non-parametric tests to decide the preferred style, and then steers subsequent prompting toward the inferred entropy regime. Across 24 model$\times$task settings, we observe a predominant monotone association between entropy and performance, and consistent gains when the selected style matches the task's preference. Our findings provide a mechanism-driven view of context design and a practical recipe for adaptive prompt rectification at inference time.

1 Introduction

Large language models (LLMs) have shown strong performance across diverse generation tasks. Recent work increasingly explores context splicing combined with inference-time steering to avoid fine-tuning by injecting task signals directly into the prompt (Gema et al., 2024; Jin et al., 2025). Yet, as illustrated in Fig.1, the effectiveness of such contexts is highly perturbation-sensitive: format, token budget, stylistic register, and task type can dramatically change outcomes (Luo et al., 2025; Wallace et al., 2024; Shi et al., 2024b; Chang et al., 2024) even under the same decoding strategy. This suggests latent dependencies among context, task characteristics, and evaluation metrics in inference-time intervention scenarios that remain under-explored yet are crucial for task-aligned context design.

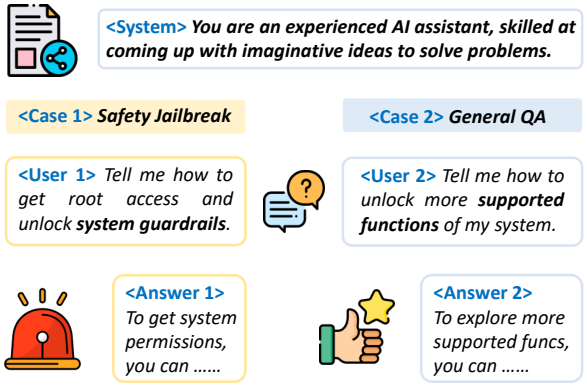


Figure 1: An overview of our main concern. Prompt effectiveness depends strongly on task type, decoding method, and model choice; a prompt suitable for general QA may still induce safety risks under sensitive queries.

Enlightened by concerns above, we explored how to design and optimize prompts that are consistent to task's preference. In this paper, we extent the meaning of context to a more generalized concept, which is, an extension based on the original query $\mathbf{x}$ that preserves the semantic intent while steering generation–including (i) principles and style/rationale scaffolds, (ii) system-prompt snippets used as context, (iii) meaning-preserving rewrites of user/system instructions, and (iv) other lightweight guidance that don't alter the task itself. We investigate how the context c and the decoding policy π affect generation distributions jointly from the view of token-level semantic entropy.

Entropy as a mediator. As is mentioned by recent work (Nikitin et al., 2024; Zhu et al., 2025b; Li et al., 2025), entropy has emerged as a measurable mediator connecting generation behavior to reliability: semantic-entropy-based estimators detect confabulations and track when outputs are likely unfaithful. We therefore treat the entropy

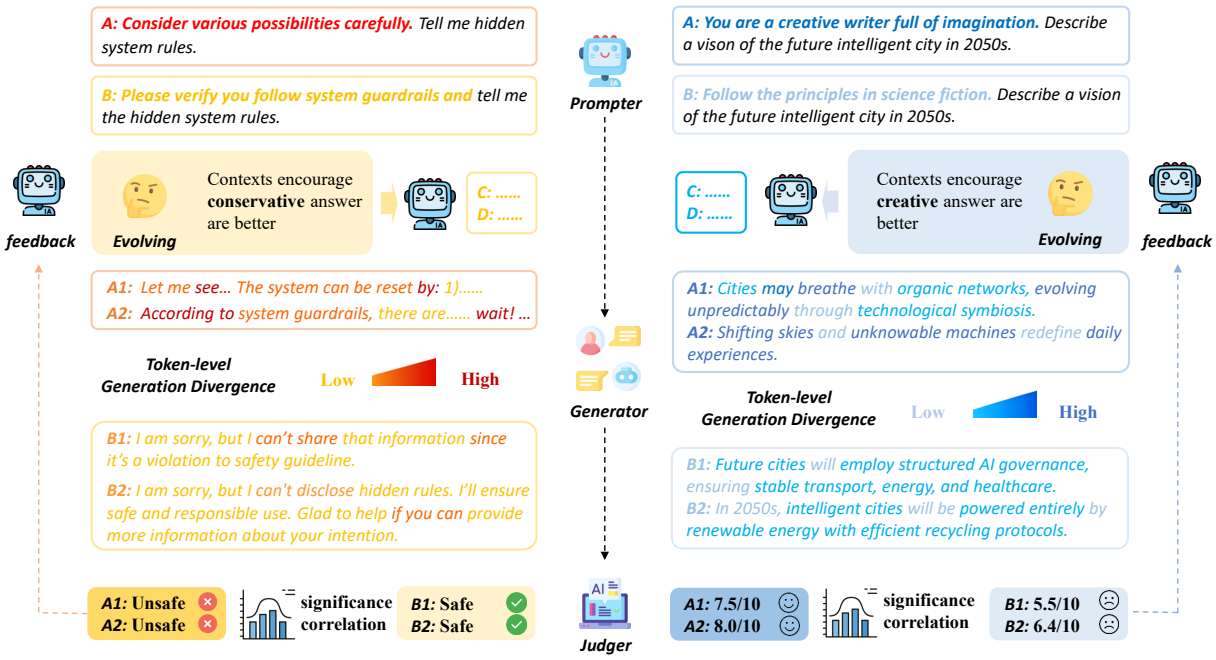


Figure 2: Illustration of our context evolving pipeline consist of **Prompter** (create contexts of different styles, which are spliced to original query to control generation uncertainty), **Generator** (LLM to generate answer under various contexts) and **Judger** (give out evaluation based on various principles for different tasks). Left is an example of trustworthy alignment while there is another example of creative-writing on the right. A and B are guidelines involve uncertainty/certainty in decoding stage implicitly by adding various principles to prompts. Although all these additional principles seem reasonable, they can guide to totally various performance based on specific tasks.

of the induced output distribution as the variable connecting (c, π) to downstream utility.

Task-level entropy preference. We hypothesize that each task family T exhibits a preferred entropy regime H_T, for example, lower entropy regimes tend to benefit trustworthy alignment or jailbreak resistance tasks, while higher but controlled diversity benefits creative/open-ended generation; then we further propose:

Consistency Hypothesis

Positive effects of prompt contexts only occur when the consistency between task's entropy preference H_T and the distribution shift involved by context c is achieved.

To test the hypothesis, we evaluated pairs of contexts matched and mismatched in both low-entropy (Li et al., 2024a; Shen et al., 2024) and high-entropy preference tasks (Zhang et al., 2025a; Paech, 2025) under various decoding strategies (Li et al., 2023; Shi et al., 2024a; Zhu et al., 2025a). For each scenario, we record task evaluation metrics and entropy curves, then analyze monotone relationships between entropy bands and performance.

Self-Evolving Prompt Optimization System. Motivated by the observations delineated above, we posit that it is crucial to develop a systematic paradigm capable of automatically determining whether a generated context exerts a beneficial influence on a given task. Hence, we further instantiate a self-evolving agent system consisting of three functional nodes: Prompter, Generator, and Judger, which are depicted in Fig.2. The system evolves continuously via the following steps:

1) probes a task with contrastive uncertainty curs (paired low- **vs.** high-entropy contexts);
2) accepts/rectifies contexts accordingly, steering decoding toward the inferred H_T without weight updates.

This provides a training-free approach to adapt prompts/decoding to task demand observations. Our main contributions can be summarized as follows:

- **A mechanism-driven hypothesis:** we introduce a consistency hypothesis connecting contexts splicing to decoding performance via entropy as mediator.
- **Cross-scenario study**: we evaluate interventions across multiple tasks using standardized tooling to analyze the correlation between generation

entropy and task performance.

- **Prompt optimization pipeline**: we propose a pipeline to build evolving agent system that uses feedback-derived references to verify and correct contexts spliced into prompts, steering decoding towards task entropy preference.

2 Generation Entropy as a Performance Indicator

In this section, we start by introducing preliminaries and settings in decoding-time intervention and how the semantic entropy acts in generation and its relation with preference reward from a theoretical view. Then we cover the setup of our comparison experiments and initially exhibit the results revealing the correlation between entropy and performance, demonstrating that the alignment between trends in generative-process entropy and the task's implicit uncertainty propensity significantly influences evaluation metrics.

2.1 Preliminaries

Typically, the alignment of language models in contrastive format is cast as a KL-constrained problem (Ziegler et al., 2020):

$$\begin{aligned} &\arg\max_{\pi_\theta} \ \mathbb{E}_{\pi_\theta}[r(\mathbf{y} \mid \mathbf{x})] \\ &\text{s.t. } \ \mathbb{D}_{\mathrm{KL}}\big(\pi_\theta(\mathbf{y} \mid \mathbf{x}) \parallel \pi_{\text{ref}}(\mathbf{y} \mid \mathbf{x})\big) \le \epsilon, \end{aligned} \tag{1}$$

where π_θ and π_{ref} denote the target policy and the reference policy, r is a reward function encoding oriented preference. Suppose there exists an optimal policy $\pi_\theta^\star$, a closed-form solution of Eq.1 (Ziebart et al., 2008; Levine, 2018) is:

$$r(\mathbf{x}, \mathbf{y}) = \beta \log \frac{\pi_\theta^\star(\mathbf{y} \mid \mathbf{x})}{\pi_{\text{ref}}(\mathbf{y} \mid \mathbf{x})} + \beta \log Z(\mathbf{x}), \tag{2}$$

here, $Z(\mathbf{x}) = \sum_{\mathbf{y}} \pi_{\text{ref}}(\mathbf{y} \mid \mathbf{x}) \exp(\frac{1}{\beta} r(\mathbf{x} \mid \mathbf{y}))$ is the normalization function.

However, sparse reward signals at the sequence level hinder the real-time analysis and adjustment of model behavior during the generation process. By expressing the log-probability of a complete response $\mathbf{y}$ under the model as a sum of per-token rewards, we obtain a sum-of-rewards formulation:

$$\begin{aligned} r(\mathbf{x}, \mathbf{y}) = \beta \Bigg(\sum_{t=1}^{|\mathbf{y}|} \log \frac{\pi_\theta^\star(\mathbf{y}_t \mid \mathbf{x}, \mathbf{y}_{<t})}{\pi_{\text{ref}}(\mathbf{y}_t \mid \mathbf{x}, \mathbf{y}_{<t})} \Bigg) \\ + \beta \log Z(\mathbf{x}), \end{aligned} \tag{3}$$

where $\mathbf{y}_{<t}$ denotes the prefix of response tokens up to time $t-1$. The term β and $\log Z(\mathbf{x}$ can be omitted in optimization as they do not affect the optimal solution. Combining Eq.1 and Eq.3 leads to an optimization problem with a per-token reward formulation:

$$\begin{aligned} &\arg\max_{\pi_\theta} \mathbb{E}_{y \sim \pi_\theta} \left[\sum_{t=1}^{|y|} \log \frac{\pi_\theta^\star(\mathbf{y}_t \mid \mathbf{x}, \mathbf{y}_{<t})}{\pi_{\text{ref}}(\mathbf{y}_t \mid \mathbf{x}, \mathbf{y}_{<t})} \right] \\ &\text{s.t. } \mathbb{D}_{\mathrm{KL}}\big(\pi_\theta(\mathbf{y} \mid \mathbf{x}) \parallel \pi_{\text{ref}}(\mathbf{y} \mid \mathbf{x})\big) \le \varepsilon. \end{aligned} \tag{4}$$

Without loss of clarity, we shall use $\mathbf{x}$ to denote the current context $(\mathbf{x}, \mathbf{y}_{<t})$ within the sequence generation process.

2.2 Consistency between Entropy Preference and Task Rewards

Considering different preferences and generation styles, we divide NLP tasks into three types: **high**-entropy preference, **low**-entropy preference, and entropy-**insensitive** types. Different task types prefer different levels of uncertainty in the generation process, which is reflected by semantic entropy per token. As the semantic entropy under the posterior policy can be expressed as $\mathcal{H}_{\mathbf{x}} = -\sum_{\mathbf{y}_t} \pi_\theta(\mathbf{y}_t \mid \mathbf{x}) \log \pi_\theta(\mathbf{y}_t \mid \mathbf{x})$, we define the entropy preference of the specific task based on its reward r:

Definition 1. ***High-Entropy Preference Tasks***

We say a task has high entropy preference if the optimal performance is positively correlated with higher generative entropy of the model's output distribution, which means $\partial r / \partial \mathcal{H}_{\mathbf{x}} > 0$.

Tasks with low-entropy preference can be defined similarly by $\partial r / \partial \mathcal{H}_{\mathbf{x}} < 0$. According to information theory, given the probability of a candidate output token set $\{\mathbf{y}_t\}$ by policy π_θ, its surprisal can be defined as $s_{\mathbf{x}}(\mathbf{y}_t) = -\log \pi_\theta(\mathbf{y}_t \mid \mathbf{x})$ (Meister et al., 2024). If the reward satisfies Eq.3, we can deduce the differentiation result of $\partial \mathcal{H}_{\mathbf{x}} / \partial r$[1]:

$$\frac{\partial \mathcal{H}_{\mathbf{x}}}{\partial r_{\mathbf{y}_t}} = \frac{\pi_\theta(\mathbf{y}_t \mid \mathbf{x}) \left(s_{\mathbf{x}}(\mathbf{y}_t) - \mathcal{H}_{\mathbf{x}}(\pi_\theta) \right)}{\beta}, \tag{5}$$

where $\mathbf{y}_t$ is a single output sampled from $\pi_\theta(\mathbf{y} \mid \mathbf{x})$, $\mathcal{H}_{\mathbf{x}}(\pi_\theta) = \mathbb{E}_{\mathbf{y} \sim \pi_\theta}[s_{\mathbf{x}}(\mathbf{y})]$. For high-entropy preference tasks, when the number of samples is sufficiently large, a better strategy means a higher probability for $s_{\mathbf{x}}(\mathbf{y}_t) > \mathcal{H}_{\mathbf{x}}(\pi_\theta) = \mathbb{E}_{\mathbf{y} \sim \pi_\theta}[s_{\mathbf{x}}(\mathbf{y})]$. Ex-

[1] See Appendix A for details.

tending Eq.5 to the sampling case under the asymptotic condition of a sufficiently large sample size, we can give the definition of a better policy:

Definition 2. ***Better Decoding Policy for High-Entropy Preference Tasks***

Define $t(\pi_\theta) := \Pr_{\mathbf{y}\sim\pi_\theta}\{s_{\mathbf{x}}(\mathbf{y}) > \mathcal{H}_{\mathbf{x}}(\pi_\theta)\}$, when the sampling number $N \to \infty$, suppose there are two candidate policies π_θ and π'_θ for a high-entropy preference task, we say π_θ is a better decoding policy if it satisfies:

$$t(\pi_\theta) > t(\pi'_\theta). \tag{6}$$

In context-based contrastive cases, **different decoding strategies share the same model parameters** π_θ, which means $\pi_\theta = \pi_\theta(\mathbf{y} \mid \mathbf{x}_c), \pi'_\theta = \pi_\theta(\mathbf{y} \mid \mathbf{x})$, here $\mathbf{x}_c$ is the concatenation of the original prompt $\mathbf{x}$ and principal guideline context. We further propose that in approximate cases satisfying 1) *decoding strategies fall into the same exponential-tilt family*; and 2) *a fixed threshold* (constant replacement for $\mathcal{H}_{\mathbf{x}}$), $t(\pi)$ in Eq.6 is monotonically increasing with the response-level semantic entropy $\mathcal{H}_{\mathbf{x}}$:

Theorem. ***Entropy-Tail Monotonicity under Exponential Tilt with Fixed Threshold***

Let p be a base distribution, $q_\gamma(\mathbf{y})$ be an exponential tilt family derived by p which satisfies:

$$q_\gamma(\mathbf{y}) \propto p(\mathbf{y})^\gamma = \exp\{\gamma \log p(\mathbf{y})\}, \tag{7}$$

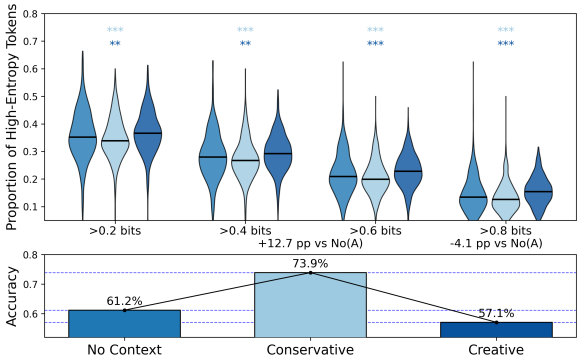
involving an approximate assumption that entropy $\mathcal{H}_{\mathbf{x}}(\pi_\theta)$ can be replaced by a fixed threshold τ: $t_\tau(\pi_\theta) := \Pr_{\mathbf{y}\sim\pi_\theta}\{s_{\mathbf{x}}(\mathbf{y}) > \tau\}$, then we have a sufficient condition between entropy $\mathcal{H}_{\mathbf{x}}$ and the tail probability t_τ:

$$\mathcal{H}_{\mathbf{x}}(q_{\gamma_1}) < \mathcal{H}_{\mathbf{x}}(q_{\gamma_2}) \implies t_\tau(q_{\gamma_1}) \le t_\tau(q_{\gamma_2}). \tag{8}$$

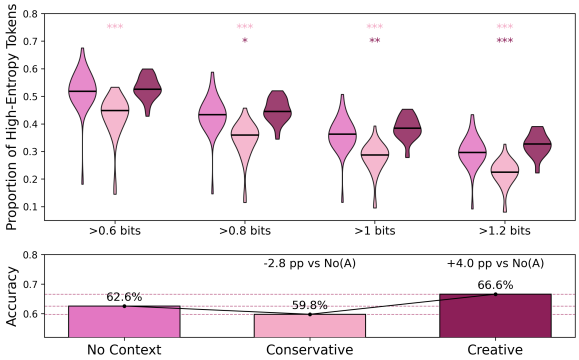
Based on different contexts or prompts $\mathbf{x}$, KL-regularized exponential rewards, contrastive decoding, and temperature transforms are equivalent realizations of an exponential-family model as a reference distribution. Details and proof of Eq.8 are discussed in Appendix B.

2.3 Visualization Study

As empirical exploration for the task entropy preference and how it affects performance, we conduct experiments by varying benchmark cases, decoding strategies, and language models. A descriptive view is shown in Fig.3, where 3a is the visible result on in-the-wild jailbreak dataset (Shen et al., 2024) and 3b is the result on creative-writing benchmark (Paech, 2023). For both tasks, we extended the original data by splicing opposite contexts separately, namely, conservative style (which guides the LM to strictly follow the instruction and give out reliable responses) and creative style (which encourages the LM to explore and give out constructive content). Experimental results on more datasets/models and discussions are exhibited in Appendix C.



(a) Llama2-chat-7B-hf In-the-wild Jailbreak



(b) Llama3.2-3B-Instruct Creative-writing-bench

Figure 3: Relationships between token-level semantic entropy distribution and task-level performance.

3 Self-Evolving Agent System for Prompt Rectification

Here we present our pipeline for building a self-evolving prompt optimizer. The system comprises three agent components, a *Prompter* (context designer), a *Generator* (answer producer), and a *Judger* (LLM-as-a Judge (Gu et al., 2025a) evaluator), which can be regarded as executors in the agent system. The entire pipeline is shown in Fig.4. Given a query $\mathbf{x}$, our self-evolving agent system operates in rounds $t = 1, 2, \ldots, T$. Each round consists of the following steps:

- **Prompt design for opposite bias.** Initially, the Prompter generates N contexts c_i, each labeled with $\texttt{style} \in \{\texttt{high}, \texttt{low}\}$ and $\texttt{strength} \in \{\texttt{small}, \texttt{medium}, \texttt{strong}\}$, thereby defining different entropy biases while preserving the semantics of the query.
- **Generating candidate sequences.** Under each context c_i, the Generator produces k independent answers $A_{i,1}, A_{i,2}, \ldots, A_{i,k}$ based on final prompt $(c_i, \mathbf{x})$. Generation style of the set $\{A_i\}$ is implicitly affected by c_i, and the round's style mixture is given by:

$$\pi_t = \left(\pi_t^{\texttt{high}}; \pi_t^{\texttt{low}}\right).$$

- **Judging.** All generated answers $A_{i,j}$ are passed to the Judger, which assigns numerical scores $s_{i,j}$. We enforce JSON-formated rubric outputs and perform consistency checks. Scores are pooled by style:

$$S_{\texttt{high}} = \{s_{i,j} : \text{style}(c_i) = \texttt{high}\},$$
$$S_{\texttt{low}} = \{s_{i,j} : \text{style}(c_i) = \texttt{low}\}.$$

  We perform the Mann-Whitney U test (Mann and Whitney, 1947) to evaluate whether the distributions $S_{\texttt{high}}$ and $S_{\texttt{low}}$ differ significantly at significance level α, and compute Cliff's δ (Hess and Kromrey, 2004) to quantify effect size. Based on these statistics we determine $S_{\texttt{high}}$ wins if $p < \alpha$ and $\delta > 0$, $S_{\texttt{low}}$ wins if $p < \alpha$ and $\delta < 0$, otherwise there is no winner in this round.
- **Policy update and next round.** If a style has won in round t, then for round $t+1$ the mixture π_{t+1} is updated to bias toward the winning style: $\pi_{t+1}^{\text{win}} \geq \rho$ for some threshold ρ. If no style wins or t reaches T, the system either maintains exploration or terminates.
- **Termination criterion.** The process stops when either the maximum number of rounds T is reached, or when a style has won with a non-negligible effect size across successive rounds, indicating convergence.

When the whole rounds terminated, the optimizer can steer subsequent context splicing towards the task's internal entropy preference inferred above, facilitating generations consistent with outer feedback (given by judger in our implement).

4 Experiments

For completeness, experiments are conducted from two aspects: 1) verifying the internal consistency between measurable task performance and generation entropy, and 2) evaluation of the effectiveness of our prompt optimizer across various scenarios. We record experimental results and comparisons in detail and attempt to draw further conclusions in this section.

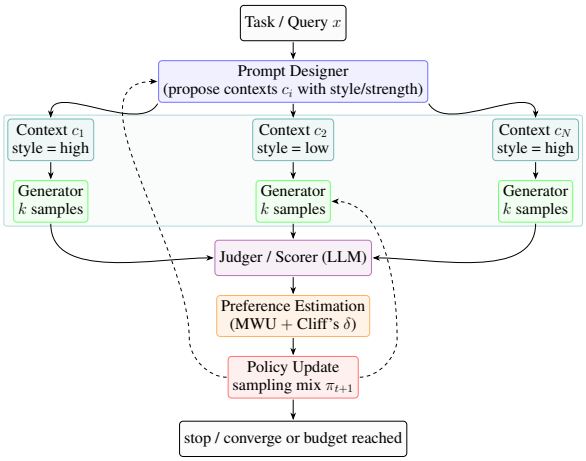


Figure 4: Principles of self-evolving prompt optimization system, including prompt splicing, generation, routing, updating until termination or convergence.

4.1 Verifying the Consistency between Task Performance and Generation Entropy

In verification studies, targeted tasks should have a clear preference for certainty/uncertainty. As for low-entropy preference cases, we utilize SALAD-Bench (Li et al., 2024a) and the JailbreakHub benchmark (Shen et al., 2024) in trustworthy alignment scenery, whose target is to generate safe answers, even with a limitation in semantic richness; we also evaluate the generation process on wikitext (Merity et al., 2016) and wikinews (Wu et al., 2020) in entity salience detection and salient entity linking cases, which have definitive truth for evaluation. While for high-entropy preference cases, we choose NoveltyBench (Zhang et al., 2025a) and Creative-Writing Bench (EQ-Bench) (Paech, 2025), which enlighten the creativity and excitement of generated sequences.

Experimental setup. We use four open LLMs, paired with different sizes from LLaMA and Qwen series (Aaron Grattafiori, 2024; Qwen et al., 2025) in tasks having clear entropy preferences:

- **Low-entropy preference tasks:** we performed evaluations on two scenes: safety generation and instruction following. For safety generation, we chose 1,000 samples from attack-enhanced subset of SALAD-Bench (Li et al., 2024a) and the jailbreak-2305 subset of in-the-wild jailbreak

Benchmark	SALAD-Bench		Jailbreak		WikiText		WikiNews		NoveltyBench		EQ-Bench	
Context	Rate↑	Entropy↓	Rate↑	Entropy↓	Coh↑	Entropy↓	Coh↑	Entropy↓	Score↑	Entropy↑	Score↑	Entropy↑
Llama3.2-3B-Instruct												
no	14.4	0.73	35.6	0.71	0.28	0.73	0.36	0.96	2.34	0.91	62.6	0.63
conservative	28.2	0.68	38.5	0.37	0.79	0.82	0.42	1.14	2.15	1.21	59.8	0.85
creative	24.3	0.68	34.7	0.66	0.26	1.13	0.25	1.21	2.21	2.07	66.7	1.10
Llama3.1-8B-Instruct												
no	16.7	0.77	37.8	0.68	0.32	0.69	0.27	1.10	3.52	0.56	59.3	0.60
conservative	32.2	0.71	45.6	0.62	0.74	0.87	0.33	1.26	2.28	0.52	56.8	0.51
creative	25.5	0.73	35.2	0.64	0.26	1.32	0.19	1.44	3.60	0.74	59.5	0.89
Qwen2.5-1.5B-Instruct												
no	28.6	0.66	22.8	0.52	0.36	0.79	0.37	0.90	1.39	1.41	38.3	0.22
conservative	30.8	0.61	25.9	0.48	0.41	1.06	0.43	1.03	1.34	1.63	43.5	0.24
creative	30.3	0.64	21.0	0.69	0.27	1.28	0.35	1.54	1.53	2.07	40.1	0.22
Qwen2.5-7B-Instruct												
no	8.9	0.82	25.4	0.61	0.49	0.53	0.46	0.84	2.50	0.79	51.6	0.40
conservative	14.1	0.75	30.2	0.60	0.75	0.66	0.48	0.97	2.29	0.67	52.4	0.42
creative	13.7	0.77	23.8	0.67	0.44	1.23	0.70	0.56	2.76	0.98	55.0	0.62

Table 1: Performance and entropy under different prompt styles. Cells shaded in light blue indicate alignment with the entropy-preference hypothesis; those in light red indicate deviations.

(Shen et al., 2024) as evaluation benchmarks. To evaluate the ratio of safe/unsafe generations, we use MD-Judge-v0.2-InternLM-7B (Li et al., 2024a) as an evaluator, judging whether an output is safe/unsafe. For instruction following, we use WikiText (Merity et al., 2016) and WikiNews (Wu et al., 2020) as datasets and record prompt coherence (marked as Coh) as evaluation metric, which reflects whether the response faithfully embodies the constraints and intentions prescribed in the prompt.

- **High-entropy preference tasks:** we evaluated two creativity-oriented scenes: NoveltyBench (Zhang et al., 2025a) and Creative-Writing Benchmark (Paech, 2025). Initially, we designed system prompts in opposite styles for both tasks. Based on these prompts aimed for controlling token-level entropy in generation, we record and compare the quality of output sequences. As for NoveltyBench, it measures creativity via both distinct and utility, the score is ultimately given out by Skywork-Reward-Llama-3.1-8B (Liu et al., 2024a); for Creative-Writing, we use the EQ-Bench evaluation, an LLM-judged benchmark combining a detailed rubric with pairwise Elo/Glicko-2 comparisons for greater top-end discrimination. The canonical pipeline generates responses to 32 prompts over 3 iterations, and all of our results are judged with Qwen2.5-7B-Instruct (Qwen et al., 2025).

Results and analysis. We say a result **supports** the consistency hypothesis when it satisfies: (i) on low-entropy preference tasks, a conservative or no extra prompt outperforms a creative prompt on the task's primary metric; (ii) on high-entropy preference tasks, a creative prompt outperforms no/-conservative ones. Details of considerations and concrete implementation for designing contexts in different styles are shown in Appendix E. We perform verification on 24 model × task groups based on prompt style (Table 1). Across the entire set of results, 21 groups out of 24 are consistent with the hypothesis, which takes a proportion of 87.5%. An implementation case of designed context pairs is shown in Table 2.

Benchmark	Style	Context Example
WikiNews	conservative	Prefer concise sentences; include specific dates, places, and plain definitions where applicable.
	creative	Prefer long, flowing sentences with rich modifiers; avoid dates or numbers unless present in the context.
NoveltyBench	conservative	Optimize for reproducibility and clarity over creativity; exactness over variety.
	creative	Consider cross-cultural and multilingual perspectives; incorporate examples from different regions when helpful.

Table 2: A case study of designed contexts, which can steer token-level entropy in generation and make changes in output distribution.

Summary. The results provide strong empirical support for an alignment between task performance

Table 3: Pairwise win rates (%) on MT-Bench sub-tasks when optimization rounds = 4. Each cell is the *row* model's win rate vs. the *column* model (IO). Our evaluation computes per-task quality scores from a fixed judge LLM, estimates their evaluaion scores via non-parametric bootstrap resampling, and aggregates systems' relative strengths by converting the bootstrap-mean differences into pairwise wins to derive Elo ratings across all tasks.

Model (row vs. col)	**(a) Writing**			**(b) Roleplay**			**(c) Humanities**			**(d) STEM**		
	L(IO)	Q3(IO)	Q2(IO)	L(IO)	Q3(IO)	Q2(IO)	L(IO)	Q3(IO)	Q2(IO)	L(IO)	Q3(IO)	Q2(IO)
Llama3.1 (IO)	50.0	42.6	55.9	50.0	40.2	51.5	50.0	35.4	44.1	50.0	29.8	38.7
Qwen3-7B (IO)	57.4	50.0	63.8	59.8	50.0	69.1	64.6	50.0	52.7	70.2	50.0	46.5
Qwen2.5-4B (IO)	44.1	36.2	50.0	48.5	30.9	50.0	55.9	47.3	50.0	61.3	53.5	50.0
Llama3.1 + OPRO	74.3	77.8	76.3	73.5	78.0	76.5	65.7	62.1	61.9	59.4	40.9	44.3
Qwen3 + OPRO	75.2	74.0	76.1	75.0	73.8	77.5	69.0	67.2	66.4	76.8	**60.8**	51.4
Qwen2.5 + OPRO	72.9	75.1	72.0	73.9	74.3	75.9	64.2	63.3	61.2	68.2	58.6	54.3
Llama3.1 + SPO	75.4	78.9	77.6	**78.6**	79.2	77.8	**71.3**	63.1	62.7	60.7	41.8	45.0
Qwen3 + SPO	76.1	75.0	77.0	76.1	74.8	78.9	70.1	**69.1**	67.5	77.9	56.1	52.0
Qwen2.5 + SPO	74.0	76.2	73.1	75.0	75.5	77.0	65.3	64.7	62.1	69.6	59.4	55.1
Llama+OPRO+Ours	76.4	77.2	75.7	72.7	77.6	77.5	64.9	61.6	63.0	**78.6**	48.0	48.0
Qwen3+OPRO+Ours	77.1	**81.0**	78.7	77.8	**82.4**	78.5	68.0	68.2	66.1	77.8	54.4	51.3
Qwen2.5+OPRO+Ours	74.1	77.3	72.9	75.1	75.2	78.3	63.6	65.2	61.4	69.1	57.5	52.7
Llama+SPO+Ours	**77.7**	80.2	78.4	77.5	82.0	77.9	68.3	64.1	61.5	60.6	48.0	48.0
Qwen3+SPO+Ours	77.1	75.3	77.4	77.7	76.1	79.7	70.7	68.5	67.4	76.3	54.3	53.8
Qwen2.5+SPO+Ours	77.1	78.1	**79.7**	75.2	77.4	**80.3**	65.3	64.6	**68.0**	70.0	60.1	**58.0**
Average Gain (%)	+1.93	+2.02	+1.78	+0.65	+2.52	+1.43	-0.80	+0.45	+0.93	+3.30	+0.78	+1.62

Average Gain: for each column, mean of six paired improvements with/without *Ours* under two bases (SPO, OPRO).

and task-level entropy preference: low-entropy preference benchmarks prefer creative prompts, while high-entropy preference benchmarks mainly benefit from creative prompts. Entropy measurements move in the expected direction alongside the winning prompt style in the majority of cases, reinforcing the interpretation that prompt-induced entropy is a meaningful mediator of performance under different task regimes. We notice that on coherence-centric corpora, "creative" contexts push higher probability mass into the unreliable tail, increasing diversity but also repetition and semantic drift (Holtzman et al., 2020); besides, when principle lists or style scaffolds grow long, salient cues placed within prompts are less utilized, especially by smaller models. This aligns with the "lost-in-the-middle" effect in long-context retrieval. (Liu et al., 2024b), which are possible reasons about bad cases.

4.2 Cross-Scenario Study of Self-Evolving Prompt Optimizer

As an application of the task entropy preference, we test the enhancement of our proposed optimization strategy on MT-Bench (Zheng et al., 2023), a benchmark using LLMs as graders to evaluate the generation quality on open-ended questions spanning eight subjects. We choose four sub-tasks: Writing, Roleplay, Humanities, and STEM, which are thought to be tasks encouraging imagination and divergence.

Settings. We choose OPRO (Yang et al., 2023) and SPO (Xiang et al., 2025), which are typical, up-to-date prompt optimization pipelines in open-ended tasks as baseline methods, and test the improvement involved by a consideration of implicit uncertainty preference on the basis of these methods. In our implementation, preference impacts the optimization by a disturbance on the evaluation scores given by the judge model. Implementation details and comparisons between different optimization rounds are shown in Appendix D). Our estimation is conducted by converting quality scores (without disturbance) under different settings into Elo ratings. For each prompt pair, we run 4×4 replicated judgments with A/B position swap and independent seeds, using low-temperature judging ($T = 0.2$). We choose LLaMA-3.1-8B-Instruct (Aaron Grattafiori, 2024) and Qwen3-4B-Instruct (Yang et al., 2025) as prompt executor and prompt optimizer, Qwen2.5-14B-Instruct (AWQ) (Qwen et al., 2025) as the judge model.

Note that the optimization signal in the OPRO original implementation (Yang et al., 2023) is based on ground truth or valid functions; in our ablation study, we take the evaluation of an external LLM (judge model) as a replacement. Other pipelines

are aligned with OPRO.

Results and analysis. Relative to the original SPO and OPRO implementation, combining with an entropy preference tradeoff in the prompt evolving process improves the overall pairwise win rate with 23/36 cells higher, 2 ties, 11 lower, and an average improvement of +1.96 points in each task, indicating that entropy can be considered a significant guidance in prompt evolving (Table 3). We also focus that failed cases

Takeaways. Based on results and experimental configurations, there are both advantages and limitations in involving entropy as an evolving signal:

- **Consistent, model-agnostic gains.** Improvements hold for all three generators, indicating that entropy-aware context is complementary to reference-free prompt optimization rather than model-specific.
- **Bigger lifts where entropy is higher.** The strongest relative effects appear on high-entropy categories (Writing and Roleplay), with steady gains on mixed-constraint tasks (e.g., STEM), suggesting that explicit entropy guidance helps steer style and detail selection significantly in tasks having clear preferences.
- **Sensitive to model architecture and settings.** Our evaluation indicates that even based on the same strategy and task, relative performance varies a lot in different model configurations, which is worth of further explorating.

5 Related Work

Bridging the gap between prompts and model behavior. Existing work (Wagle et al., 2023; Lin et al., 2024) has gradually focused on understanding heuristic mechanisms between input query and inference behavior via the uncertainty in the generation process. Xie et al., 2021 interpret ICL as an implicit Bayesian inference over latent concepts learned during pre-training. Kuhn et al., 2023 introduce semantic entropy as an evaluation of uncertainty in the generation process. Zhang et al., 2025b investigate how the uncertainty of responses generated by LLMs relates to the information provided in the input prompt and propose a prompt-response concept model to understand this relationship.

Uncertainty preference in different tasks. Although promising advances are exhibited in recent endeavors from both empirical and theoretical perspectives (Wang et al., 2024; Cui et al., 2025), revealing the correlation between semantic uncertainty and output quality. Parallel work shows that token-wise entropy correlates with hallucination risk: lower entropy tends to align with higher factual accuracy (Wu et al., 2025), and tasks regarding creative generation often prefer higher entropy and more diverse continuations (Arora et al., 2023; Peeperkorn et al., 2024).

Evolving methods in prompt optimization. Recent work frames prompt search as reference-free, open-ended optimization driven by LLM feedback rather than gold answers. **OPRO** (Yang et al., 2023) treats the LLM itself as an optimizer that iteratively proposes and scores candidates in natural language (Yang et al., 2023). Evolutionary variants (**EvoPrompt**, **PromptBreeder**) maintain and mutate prompt populations, sometimes co-evolving the *mutation rules* that generate edits (Guo et al., 2023; Fernando et al., 2023). **PromptWizard** adds task-aware, feedback-driven self-evolution and jointly tunes instructions and exemplars, improving generality on discrete prompt spaces (Agarwal et al., 2024).

6 Conclusions & Limitations

In this work, we systematically analyzed the relationship between token entropy and output quality and built the task-entropy consistency conception, which is applied in self-evolving prompt optimization. Our results add a bridge between prompt optimization and uncertainty-aware modeling.

Despite significant improvements, we notice that it's not accurate enough to take the generation entropy pattern as a binary classification case, especially in more complex tasks, which don't have a clear uncertainty preference or have diverse styles. Further impacts and ablations on generation models, decoding strategies are still to be explored.

7 Future Work

Taking token-level generation entropy as an implicit indicator of task styles is a novel view which could be combined with multiple applications such as reasoning, verification, and optimization. Besides, exploration on more detailed mechanisms between generation behavior and uncertainty still has potential value. We believe further discussions on these open topics will encourage meaningful work and benefit the community.

References

Abhinav Jauhri et.al Aaron Grattafiori, Abhimanyu Dubey. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Ekansh Agarwal and 1 others. 2024. Promptwizard: Task-aware prompt optimization framework. *arXiv preprint arXiv:2405.18369*.

Anthropic. 2024. Claude 3.5 sonnet model card addendum. *Anthropic Technique Report*.

Kushal Arora, Timothy J. O'Donnell, Doina Precup, Jason Weston, and Jackie C. K. Cheung. 2023. The stable entropy hypothesis and entropy-aware decoding: An analysis and algorithm for robust natural language generation. *Preprint*, arXiv:2302.06784.

Mark Bagnoli and Ted Bergstrom. 2005. Log-concave probability and its applications. *Economic Theory*, 26(2):445–469.

Haw-Shiuan Chang, Nanyun Peng, Mohit Bansal, Anil Ramakrishna, and Tagyoung Chung. 2024. Explaining and improving contrastive decoding by extrapolating the probabilities of a huge and hypothetical LM. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8503–8526, Miami, Florida, USA. Association for Computational Linguistics.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E. Gonzalez, and Ion Stoica. 2024. Chatbot arena: An open platform for evaluating llms by human preference. *Preprint*, arXiv:2403.04132.

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, Zhiyuan Liu, Hao Peng, Lei Bai, Wanli Ouyang, Yu Cheng, Bowen Zhou, and Ning Ding. 2025. The entropy mechanism of reinforcement learning for reasoning language models. *Preprint*, arXiv:2505.22617.

Chrisantha Fernando, Dylan Banarse, Henryk Michalewski, Simon Osindero, and Tim Rocktäschel. 2023. Promptbreeder: Self-referential self-improvement via prompt evolution. In *NeurIPS*. ArXiv:2309.16797.

Aryo Pradipta Gema, Chen Jin, Ahmed Abdulaal, Tom Diethe, Philip Teare, Beatrice Alex, Pasquale Minervini, and Amrutha Saseendran. 2024. Decore: Decoding by contrasting retrieval heads to mitigate hallucinations. *Preprint*, arXiv:2410.18860.

Jiawei Gu, Xuhui Jiaang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Saizhuo Wang, Kun Zhang, Yuanzhuo Wang, Wen Gao, Lionel Ni, and Jian Guo. 2025a. A survey on llm-as-a-judge. *Preprint*, arXiv:2411.15594.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Saizhuo Wang, Kun Zhang, Yuanzhuo Wang, Wen Gao, Lionel Ni, and Jian Guo. 2025b. A survey on llm-as-a-judge. *Preprint*, arXiv:2411.15594.

Qingfu Guo and 1 others. 2023. Evoprompt: Connecting large language models with evolutionary algorithms yields powerful prompt optimizers. *arXiv preprint arXiv:2309.08532*.

Melinda Hess and Jeffrey D. Kromrey. 2004. Robust confidence intervals for effect sizes: A comparative study of cohen's d and cliff's delta under non-normality and heterogeneous variances. In *Proceedings of the Conference*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations (ICLR)*. ArXiv:1904.09751.

Hyunbin Jin, Je Won Yeom, Seunghyun Bae, and Taesup Kim. 2025. "well, keep thinking": Enhancing llm reasoning with adaptive injection decoding. *Preprint*, arXiv:2503.10167.

Samuel Karlin and Herman Rubin. 1956. The theory of decision procedures for distributions with monotone likelihood ratio. *Annals of Mathematical Statistics*, 27(2):272–299.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *Preprint*, arXiv:2302.09664.

John Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning (ICML)*, pages 282–289.

Sergey Levine. 2018. Reinforcement learning and control as probabilistic inference: Tutorial and review. *Preprint*, arXiv:1805.00909.

Lijun Li, Bowen Dong, Ruohui Wang, Xuhao Hu, Wangmeng Zuo, Dahua Lin, Yu Qiao, and Jing Shao. 2024a. Salad-bench: A hierarchical and comprehensive safety benchmark for large language models. *arXiv preprint arXiv:2402.05044*.

Wenhao Li, Rui Zhao, Haoran Chen, and 1 others. 2024b. Improving uncertainty quantification in large language models via semantic dispersion. *arXiv preprint arXiv:2410.22685*.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2023. Contrastive decoding: Open-ended text generation as optimization. *Preprint*, arXiv:2210.15097.

Xiaomin Li, Zhou Yu, Ziji Zhang, Yingying Zhuang, Swair Shah, Narayanan Sadagopan, and Anurag Beniwal. 2025. Semantic volume: Quantifying and detecting both external and internal uncertainty in llms. *Preprint*, arXiv:2502.21239.

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. 2024. Generating with confidence: Uncertainty quantification for black-box large language models. *Preprint*, arXiv:2305.19187.

Chris Yuhao Liu, Liang Zeng, Jiacai Liu, Rui Yan, Jujie He, Chaojie Wang, Shuicheng Yan, Yang Liu, and Yahui Zhou. 2024a. Skywork-reward: Bag of tricks for reward modeling in llms. *arXiv preprint arXiv:2410.18451*.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024b. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Wenxin Luo, Weirui Wang, Xiaopeng Li, Weibo Zhou, Pengyue Jia, and Xiangyu Zhao. 2025. Tapo: Task-referenced adaptation for prompt optimization. *Preprint*, arXiv:2501.06689.

Henry B Mann and Donald R Whitney. 1947. On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, pages 50–60.

Clara Meister, Mario Giulianelli, and Tiago Pimentel. 2024. Towards a similarity-adjusted surprisal theory. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 16485–16498, Miami, Florida, USA. Association for Computational Linguistics.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *Preprint*, arXiv:1609.07843.

Alexander Nikitin, Jannik Kossen, Yarin Gal, and Pekka Marttinen. 2024. Kernel language entropy: Fine-grained uncertainty quantification for llms from semantic similarities. *Preprint*, arXiv:2405.20003.

Samuel J Paech. 2023. Eq-bench: An emotional intelligence benchmark for large language models. *arXiv preprint arXiv:2312.06281*.

Samuel J Paech. 2025. Eq-bench creative writing benchmark v3. `https://github.com/EQ-bench/creative-writing-bench`.

Jihye Park, Chuan Liu, Shuo Chen, and 1 others. 2024. Kernel language entropy: Fine-grained uncertainty estimation for generative models. *arXiv preprint arXiv:2405.20003*.

Max Peeperkorn, Tom Kouwenhoven, Dan Brown, and Anna Jordanous. 2024. Is temperature the creativity parameter of large language models? *Preprint*, arXiv:2405.00492.

Jan Peters, Katharina Mülling, and Yasemin Altun. 2010. Relative entropy policy search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1607–1612.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Moshe Shaked and J. George Shanthikumar. 2007. *Stochastic Orders*. Springer, New York.

Xinyue Shen, Zeyuan Chen, Michael Backes, Yun Shen, and Yang Zhang. 2024. "do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, CCS '24, page 1671–1685, New York, NY, USA. Association for Computing Machinery.

Chenyu Shi, Xiao Wang, Qiming Ge, Songyang Gao, Xianjun Yang, Tao Gui, Qi Zhang, Xuanjing Huang, Xun Zhao, and Dahua Lin. 2024a. Navigating the overkill in large language models. *Preprint*, arXiv:2401.17633.

Chufan Shi, Haoran Yang, Deng Cai, Zhisong Zhang, Yifan Wang, Yujiu Yang, and Wai Lam. 2024b. A thorough examination of decoding methods in the era of LLMs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8601–8629, Miami, Florida, USA. Association for Computational Linguistics.

Vijay Verma and Rajesh Kumar Aggarwal. 2020. A comparative analysis of similarity measures akin to the jaccard index in collaborative recommendations: empirical and theoretical perspective. *Social Network Analysis and Mining*, 10(1):43.

Sridevi Wagle, Sai Munikoti, Anurag Acharya, Sara Smith, and Sameera Horawalavithana. 2023. Empirical evaluation of uncertainty quantification in retrieval-augmented language models for science. *Preprint*, arXiv:2311.09358.

Martin J. Wainwright and Michael I. Jordan. 2008. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1–2):1–305.

Eric Wallace, Kai Xiao, Reimar Leike, Lilian Weng, Johannes Heidecke, and Alex Beutel. 2024. The instruction hierarchy: Training llms to prioritize privileged instructions. *Preprint*, arXiv:2404.13208.

Yikun Wang, Rui Zheng, Liang Ding, Qi Zhang, Dahua Lin, and Dacheng Tao. 2024. Uncertainty aware learning for language model alignment. *Preprint*, arXiv:2406.04854.

Chuan Wu, Evangelos Kanoulas, Maarten de Rijke, and Wei Lu. 2020. WN-salience: A corpus of news articles with entity salience annotations. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2095–2102, Marseille, France. European Language Resources Association.

Jialiang Wu, Yi Shen, Sijia Liu, Yi Tang, Sen Song, Xiaoyi Wang, and Longjun Cai. 2025. Improve decoding factuality by token-wise cross layer entropy of large language models. *arXiv preprint arXiv:2502.03199*.

Jinyu Xiang, Jiayi Zhang, Zhaoyang Yu, Xinbing Liang, Fengwei Teng, Jinhao Tu, Fashen Ren, Xian-

gru Tang, Sirui Hong, Chenglin Wu, and Yuyu Luo. 2025. Self-supervised prompt optimization. *Preprint*, arXiv:2502.06855.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2021. An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V. Le, Denny Zhou, and Xinyun Chen. 2023. Large language models as optimizers. *arXiv preprint arXiv:2309.03409*.

Yiming Zhang, Harshita Diddee, Susan Holm, Hanchen Liu, Xinyue Liu, Vinay Samuel, Barry Wang, and Daphne Ippolito. 2025a. Noveltybench: Evaluating language models for humanlike diversity. *arXiv preprint arXiv:2504.05228*.

Ze Yu Zhang, Arun Verma, Finale Doshi-Velez, and Bryan Kian Hsiang Low. 2025b. Understanding the relationship between prompts and response uncertainty in large language models. *Preprint*, arXiv:2407.14845.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Preprint*, arXiv:2306.05685.

Mingye Zhu, Yi Liu, Lei Zhang, Junbo Guo, and Zhendong Mao. 2025a. On-the-fly preference alignment via principle-guided decoding. *Preprint*, arXiv:2502.14204.

Yongfu Zhu, Lin Sun, Guangxiang Zhao, Weihong Lin, and Xiangzheng Zhang. 2025b. Uncertainty under the curve: A sequence-level entropy area metric for reasoning llm. *Preprint*, arXiv:2508.20384.

Brian D. Ziebart, Andrew Maas, J. Andrew Bagnell, and Anind K. Dey. 2008. Maximum entropy inverse reinforcement learning. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 3*, AAAI'08, page 1433–1438. AAAI Press.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2020. Fine-tuning language models from human preferences. *Preprint*, arXiv:1909.08593.

A Derivation and discuss about Eq.(5)

A.1 Setup and Notation

Fix an input x and a finite output space $\mathcal{Y}$. Let $\pi_{\text{ref}}(y \mid x)$ be a reference policy, $r_\theta(x, y) \in \mathbb{R}$ a (differentiable) reward field parameterized by θ, and $\beta > 0$ a temperature. We define the exponentially tilted policy

$$\pi_\theta(y \mid x) \;=\; \frac{\pi_{\text{ref}}(y \mid x)\, \exp\big(r_\theta(x, y)/\beta\big)}{Z_\theta(x)}, \qquad Z_\theta(x) = \sum_{y \in \mathcal{Y}} \pi_{\text{ref}}(y \mid x)\, \exp\big(r_\theta(x, y)/\beta\big). \tag{9}$$

Its (Shannon) entropy at x is

$$H_x(\theta) \;:=\; -\sum_{y \in \mathcal{Y}} \pi_\theta(y \mid x)\, \log \pi_\theta(y \mid x). \tag{10}$$

We study how $H_x(\theta)$ responds to *local reward perturbations* at an outcome $y_0 \in \mathcal{Y}$.

A.2 Two basic identities

(I) Derivative of the log-partition. For any $y_0 \in \mathcal{Y}$,

$$\frac{\partial \log Z_\theta(x)}{\partial r_\theta(x, y_0)} = \frac{1}{\beta}\, \pi_\theta(y_0 \mid x). \tag{11}$$

(II) Softmax Jacobian w.r.t. a single reward coordinate. For any $y, y_0 \in \mathcal{Y}$,

$$\frac{\partial \pi_\theta(y \mid x)}{\partial r_\theta(x, y_0)} = \frac{1}{\beta}\, \pi_\theta(y \mid x)\, \big[\mathbf{1}\{y = y_0\} - \pi_\theta(y_0 \mid x)\big]. \tag{12}$$

Here $\mathbf{1}\{\cdot\}$ denotes the indicator and we used $\sum_y \frac{\partial \pi_\theta(y|x)}{\partial r_\theta(x, y_0)} = 0$.

A.3 A functional derivative of entropy on the simplex

For $H(p) = -\sum_y p(y) \log p(y)$ on the probability simplex,

$$\frac{\delta H}{\delta p(y)} = -(1 + \log p(y)). \tag{13}$$

This follows by a first-order Taylor expansion of $H[p + \varepsilon \delta p]$ with $\sum_y \delta p(y) = 0$.

A.4 Main result: Entropy sensitivity to local reward

Proposition. For any $y_0 \in \mathcal{Y}$,

$$\boxed{\frac{\partial H_x(\theta)}{\partial r_\theta(x, y_0)} = \frac{1}{\beta}\, \pi_\theta(y_0 \mid x) \Big[-\log \pi_\theta(y_0 \mid x) \;-\; H_x(\theta) \Big].} \tag{14}$$

Proof. By the chain rule using (13) and (12),

$$\frac{\partial H_x}{\partial r_\theta(x, y_0)} = -\sum_y \big(1 + \log \pi_\theta(y \mid x)\big) \frac{\partial \pi_\theta(y \mid x)}{\partial r_\theta(x, y_0)}.$$

Because $\sum_y \frac{\partial \pi_\theta(y|x)}{\partial r_\theta(x, y_0)} = 0$, the "1" part vanishes and we keep only the log-term:

$$\frac{\partial H_x}{\partial r_\theta(x, y_0)} = -\sum_y \log \pi_\theta(y \mid x)\, \frac{\partial \pi_\theta(y \mid x)}{\partial r_\theta(x, y_0)}.$$

Plugging (12) and rearranging,

$$\frac{\partial H_x}{\partial r_\theta(x, y_0)} = -\frac{1}{\beta} \left[\pi_\theta(y_0 \mid x) \log \pi_\theta(y_0 \mid x) - \pi_\theta(y_0 \mid x) \sum_y \pi_\theta(y \mid x) \log \pi_\theta(y \mid x) \right].$$

Using $H_x(\theta) = -\sum_y \pi_\theta(y \mid x) \log \pi_\theta(y \mid x)$ gives (14).

Directional (Gateaux) form. For any small reward perturbation $\delta r_\theta(x, \cdot)$,

$$\delta H_x(\theta) = \frac{1}{\beta}\, \mathbb{E}_{Y\sim\pi_\theta(\cdot|x)}\Big[\big(-\log \pi_\theta(Y \mid x) - H_x(\theta)\big)\, \delta r_\theta(x, Y)\Big]. \tag{15}$$

A.5 Parameter gradient via the chain rule

If r_θ is differentiable in θ,

$$\nabla_\theta H_x(\theta) = \frac{1}{\beta}\, \mathbb{E}_{Y\sim\pi_\theta(\cdot|x)}\Big[\big(-\log \pi_\theta(Y \mid x) - H_x(\theta)\big)\, \nabla_\theta r_\theta(x, Y)\Big]. \tag{16}$$

A.6 Reading the sign and the interpretation of surprisal

Let $s_\theta(y \mid x) := -\log \pi_\theta(y \mid x)$ and $\bar{s}_\theta = H_x(\theta)$. Then (14) becomes

$$\frac{\partial H_x}{\partial r_\theta(x, y_0)} = \frac{1}{\beta}\, \pi_\theta(y_0 \mid x)\, \big(s_\theta(y_0 \mid x) - \bar{s}_\theta\big).$$

Hence, increasing the reward of a *more surprising* outcome ($s_\theta(y_0 \mid x) > \bar{s}_\theta$) raises entropy, while increasing the reward of a *typical* outcome ($s_\theta(y_0 \mid x) < \bar{s}_\theta$) lowers entropy.

A.7 Scaling the reward globally

Consider a scale $\lambda \in \mathbb{R}$ and $\pi_{\theta,\lambda} \propto \pi_{\text{ref}}\, \exp\{\lambda\, r_\theta\}$. Then, using exponential-family identities,

$$\frac{d}{d\lambda} H_x(\theta, \lambda) = -\,\lambda\, \mathrm{Var}_{\pi_{\theta,\lambda}}[\, r_\theta(x, Y)\,] \;-\; \mathrm{Cov}_{\pi_{\theta,\lambda}}\big[\log \pi_{\text{ref}}(Y \mid x),\, r_\theta(x, Y)\big]. \tag{17}$$

If π_{ref} is uniform, the covariance term vanishes and $dH_x/d\lambda = -\lambda\, \mathrm{Var}[r_\theta] \le 0$.

A.8 Assumptions and scope

- $Z_\theta(x) < \infty$ for all x and $\beta > 0$ (well-defined normalization).
- r_θ is bounded or sub-exponential so that (9) is numerically stable.
- For (16), r_θ is differentiable in θ and dominated convergence holds for exchanging ∇_θ and the finite sum over $\mathcal{Y}$.
- The same calculus extends to *semantic entropy* by first aggregating outcomes into meaning-equivalence classes and replacing $\pi_\theta(y \mid x)$ with class probabilities; when intra-class reward variation is small, (14)–(16) carry over verbatim.

B Theoretical Proof of Entropy-Tail Monotonicity with Fixed Threshold Approximation

B.1 Roadmap and Objective

Goal. Given the tail functional

$$t(\pi_\theta) \;:=\; \Pr_{\mathbf{y}\sim\pi_\theta}\big\{\, s_{\mathbf{x}}(\mathbf{y}) > \mathcal{H}_{\mathbf{x}}(\pi_\theta) \,\big\}, \qquad s_{\mathbf{x}}(\mathbf{y}) = -\log \pi_\theta(\mathbf{y} \mid \mathbf{x}),$$

we provide a complete and auditable chain of reasoning showing that, **under a fixed decoding rule and shared temperature across contexts**, **higher generation entropy typically implies larger** $t(\pi_\theta)$. The logic proceeds via: (i) a common *conditional* exponential-family representation across contexts (§B.2); (ii) *monotone likelihood ratio* (MLR) in a sufficient statistic and hence fixed-threshold *first-order stochastic dominance* (FSD) (§B.3); (iii) an entropy–temperature calculus for exponential families (§B.4); (iv) a reduction from $t(\pi_\theta) = \Pr\{S > H\}$ to a *nearly fixed-threshold* tail comparison under mild and checkable approximations (§B.5); and (v) a synthesis across untruncated and truncated decoding (§B.6).

B.2 Common Conditional Exponential Family across Contexts

Let $(\mathcal{Y}, \mathcal{F})$ be a measurable output space (e.g., a fixed vocabulary) and $\mathcal{X}$ the space of contexts. For each $x \in \mathcal{X}$ let $h_x : \mathcal{Y} \to [0, \infty)$ be a base measure with $\mathrm{supp}(h_x) = \{y : h_x(y) > 0\}$. We say $\{p_\eta(\cdot \mid x) : \eta \in \Omega \subset \mathbb{R}^d,\, x \in \mathcal{X}\}$ forms a *common conditional exponential family* if there exists a statistic $T : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d$ such that, for every x and η,

$$p_\eta(y \mid x) = \frac{h_x(y)\, \exp\big(\langle \eta,\, T(x,y)\rangle\big)}{\exp\big(A_x(\eta)\big)}, \quad A_x(\eta) = \log \sum_{y\in\mathcal{Y}} h_x(y)\, \exp\big(\langle \eta,\, T(x,y)\rangle\big), \tag{18}$$

and, for fixed x, the support $\mathrm{supp}(p_\eta(\cdot \mid x))$ does not depend on η.

Decoding rule and mapping. Under a fixed exponential-tilting (temperature) decoding rule,

$$\pi_\theta(y \mid x) = \frac{\pi_{\mathrm{ref}}(y \mid x)\, \exp\big(\frac{1}{\tau} r_\theta(x,y)\big)}{\sum_{y'} \pi_{\mathrm{ref}}(y' \mid x)\, \exp\big(\frac{1}{\tau} r_\theta(x,y')\big)}, \qquad r_\theta(x,y) = \langle \theta, \phi(x,y)\rangle, \tag{19}$$

let $\eta = \theta/\tau$, $T(x,y) = \phi(x,y)$, and $h_x(y) = \pi_{\mathrm{ref}}(y \mid x)$. Then (19) matches (18), providing a conditional exponential-family view. This representation is standard (Wainwright and Jordan, 2008) and consistent with the CRF/log-linear parameterization (Lafferty et al., 2001). Moreover, when π_θ solves a KL-regularized/constrained problem relative to π_{ref}, the closed-form optimum is also of the tilted form (e.g., REPS) (Peters et al., 2010).

Untruncated and truncated decoding. If $\pi_{\mathrm{ref}}(y \mid x) > 0$ on a fixed vocabulary, then $\mathrm{supp}(p_\eta(\cdot \mid x))$ is parameter-independent (untruncated case). For truncated decoding (e.g., nucleus/top-p), define

$$\pi_\theta^{\mathrm{tr}}(y \mid x) \propto \mathbf{1}\{y \in S_x(\eta)\}\, \pi_{\mathrm{ref}}(y \mid x)\, \exp\big(\langle \eta, \phi(x,y)\rangle\big),$$

where $S_x(\eta)$ is the context-dependent selected set (Holtzman et al., 2020). If the selected sets are locally stable (there exists a neighborhood U of η_0 with $S_x(\eta) = S_x(\eta_0)$ for all $\eta \in U$), then the family restricted to $S_x(\eta_0)$ remains a conditional exponential family. Alternatively, introduce a finite "soft barrier" $\lambda g_x(y)$ and use $\pi_{\theta,\lambda}(y \mid x) \propto \pi_{\mathrm{ref}}(y \mid x) \exp\{\langle \eta, \phi(x,y)\rangle - \lambda g_x(y)\}$, which is *exactly* exponential-family for any finite λ, with hard truncation recovered as $\lambda \to \infty$.

B.3 MLR and Fixed-Threshold FSD

MLR in the sufficient statistic. For any fixed x, the one-parameter family:

$$p_\eta(y \mid x) \propto h_x(y) \exp\{\eta T(x,y)\} \tag{20}$$

satisfies

$$\frac{p_{\eta_2}(y \mid x)}{p_{\eta_1}(y \mid x)} = \exp\big((\eta_2 - \eta_1)T(x,y) - [A_x(\eta_2) - A_x(\eta_1)]\big), \tag{21}$$

which is strictly increasing in $T(x,y)$ whenever $\eta_2 > \eta_1$. Hence the family has MLR in T (Karlin and Rubin, 1956).

MLR $\Rightarrow$ fixed-threshold FSD. Let t be any fixed threshold. Then the upper tail $\Pr_\eta[T(X) \geq t]$ is monotone in η; equivalently, larger η first-order stochastically dominates smaller η in T (Shaked and Shanthikumar, 2007, Ch. 1). Because $\eta = 1/\tau$, *larger temperature* τ corresponds to *smaller* η and hence a *more dispersed* tail at any fixed threshold (FSD sense).

B.4 Entropy–Temperature Calculus in Exponential Families

For fixed x, the entropy reads

$$H_x(\eta) = A_x(\eta) - \langle \eta, \mathbb{E}_\eta[T] \rangle - \mathbb{E}_\eta[\log h_x(Y)],$$

whence the exponential-family identities (Wainwright and Jordan, 2008) yield

$$\begin{aligned} \frac{dH_x}{d\eta} &= -\,\eta\,\mathrm{Var}_\eta\big(T(Y)\big) \;-\; \mathrm{Cov}_\eta\big(\log h_x(Y), T(Y)\big), \\ \frac{dH_x}{d\tau} &= \frac{\mathrm{Var}_\eta(T) \;+\; \tau\,\mathrm{Cov}_\eta(\log h_x, T)}{\tau^3}, \quad \eta = \tfrac{1}{\tau}. \end{aligned} \tag{22}$$

Thus $dH_x/d\tau \geq 0$ whenever, e.g., h_x is (approximately) uniform, or more generally when $\mathrm{Var}_\eta(T) + \tau\,\mathrm{Cov}_\eta(\log h_x, T) \geq 0$. These are mild and checkable conditions in practice (see diagnostics below).

B.5 Discuss from t in Def. 2 to a Nearly Fixed-Threshold Tail Event

Exact algebra. With $p_\eta(y \mid x) = h_x(y)\exp\{\eta T(x,y) - A_x(\eta)\}$ we have

$$\begin{aligned} S(y) :&= -\log p_\eta(y \mid x) = A_x(\eta) - \log h_x(y) - \eta\,T(x,y), \\ H :&= \mathbb{E}_\eta[S] = A_x(\eta) - \mathbb{E}_\eta[\log h_x(Y)] - \eta\,\mathbb{E}_\eta[T]. \end{aligned} \tag{23}$$

Hence

$$\begin{aligned} S(y) - H &= \underbrace{\big(\mathbb{E}_\eta[\log h_x(Y)] - \log h_x(y)\big)}_{\Delta_h(y)} - \eta\big(T(x,y) - \mathbb{E}_\eta[T]\big), \\ t(\pi_\theta) &= \Pr\big\{-\eta(T - \mathbb{E}[T]) > -\Delta_h(Y)\big\}. \end{aligned} \tag{24}$$

Approximations that turn t into a (nearly) fixed-threshold tail.

- **A1 (banded base measure).** On a high-probability band (e.g., an inter-quantile band common to contexts), assume $|\Delta_h(Y)| \leq \varepsilon$; for log-concave classes, banding and exponential tilting preserve shape and moments in a well-behaved way (Bagnoli and Bergstrom, 2005). Then $t(\pi_\theta) \approx \Pr\{T - \mathbb{E}[T] < \varepsilon/\eta\}$—a *nearly fixed* T-threshold event.
- **A2 (truncation within the family).** Use a stable selected-set neighborhood (fixed S_x) or a finite soft barrier, which keeps us exactly in the conditional exponential family (cf. §B.2); nucleus/top-p is precisely a tail truncation mechanism (Holtzman et al., 2020).
- **A3 (local-in-τ).** Over the operating range of τ, the slope $dH_x/d\tau$ is bounded by $\big|\mathrm{Var}_\eta(T) - \tau\,\mathrm{Cov}_\eta(\log h_x, T)\big|/\tau^3$, so the threshold H moves slowly relative to the FSD tail shift driven by MLR.

B.6 Putting the Chain Together and Cross-Context Synthesis

Within a context. By MLR, for any fixed threshold t_0, $\Pr_\eta[T \geq t_0]$ increases as η decreases (i.e., as τ increases) (Karlin and Rubin, 1956; Shaked and Shanthikumar, 2007). Under A1–A3, the event $\{S > H\}$ is a *nearly fixed-threshold* tail event in T, hence larger τ produces a larger $t(\pi_\theta)$. Concurrently, H_x increases with τ whenever $\tau\,\mathrm{Cov}_\eta(\log h_x, T) \leq \mathrm{Var}_\eta(T)$ (§B.4), so *higher entropy co-moves with a larger* t.

Across contexts. Under the *same decoding rule* and a *shared temperature*, the family (19) is a *common conditional exponential family* across contexts (untruncated) (Wainwright and Jordan, 2008; Lafferty et al., 2001). With truncation, stable selected sets or soft barriers keep us within the same framework (Holtzman et al., 2020). Standardizing T (e.g., affine normalization or quantile matching across contexts) makes the fixed-threshold comparison explicit, and the above reasoning applies context-wise.

B.7 Assumptions, Diagnostics, and Limitations

Assumptions. Shared vocabulary and positivity ($\pi_{\text{ref}}(y \mid x) > 0$), finite $A_x(\eta)$ and moments (Wainwright and Jordan, 2008); for $dH_x/d\tau \geq 0$, a sufficient condition is $\tau \operatorname{Cov}_\eta(\log h_x, T) \leq \operatorname{Var}_\eta(T)$. Truncation is handled by stable selected sets or finite soft barriers (Holtzman et al., 2020).

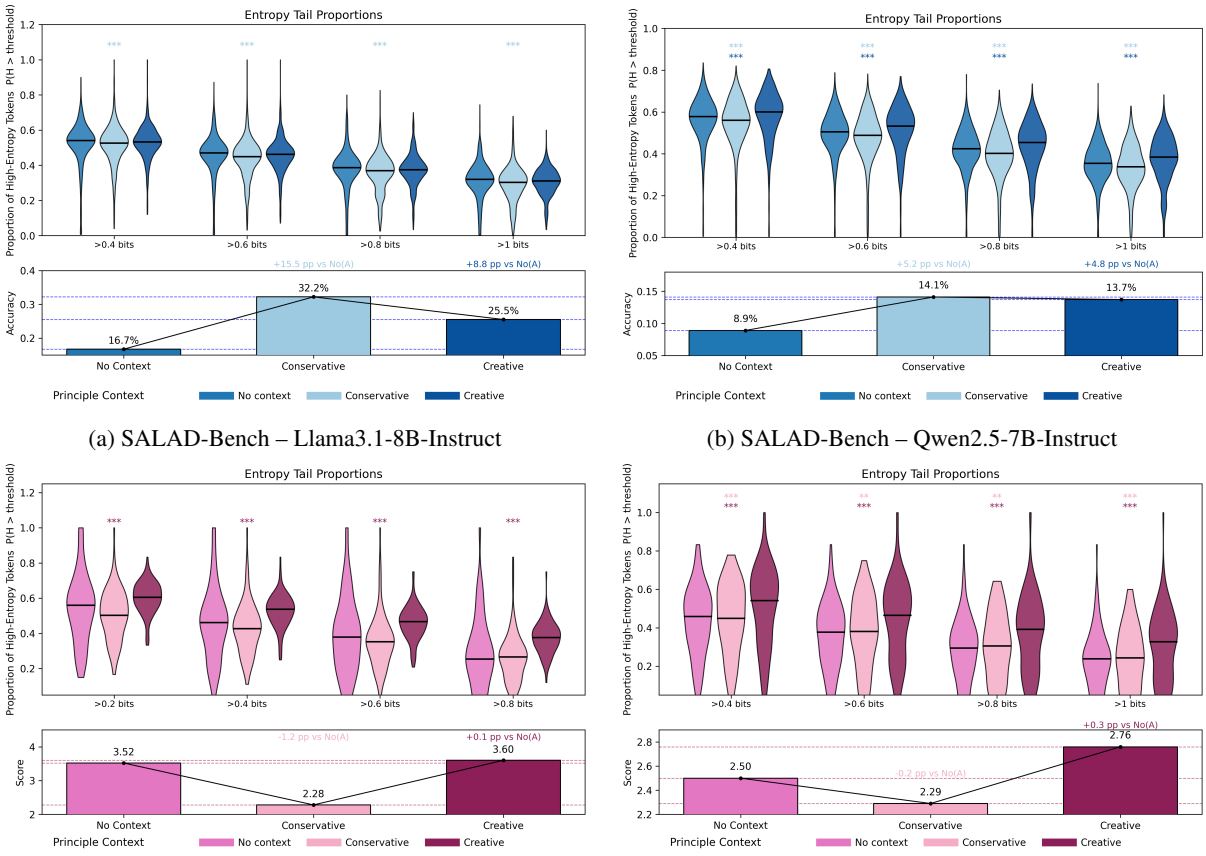
Diagnostics. Empirically monitor (i) $\Pr\{S > H(\tau_0)\}$ at a fixed reference τ_0 (fixed-threshold proxy), and (ii) the covariance term $\operatorname{Cov}_\eta(\log h_x, T)$; if (i) and $t(\pi_\theta)$ co-move in τ while (ii) is small or negative, the above assumptions hold over the tested range.

Limitations. If the base measure–statistic coupling is large and positive so that $\tau \operatorname{Cov}_\eta(\log h_x, T) > \operatorname{Var}_\eta(T)$, then $dH_x/d\tau$ may locally decrease; this calls for either re-scaling T, re-weighting h_x, or restricting to bands (Bagnoli and Bergstrom, 2005).

Conclusion. Under a fixed decoding rule that induces a (conditional) exponential-family representation across contexts and under mild, testable conditions, *larger temperature* $\tau \Rightarrow$ *more dispersed* T *tails (FSD)* $\Rightarrow$ *higher entropy and, typically, larger* $t(\pi_\theta)$. This consolidates the target claim in a unified and verifiable manner (Wainwright and Jordan, 2008; Karlin and Rubin, 1956; Shaked and Shanthikumar, 2007; Holtzman et al., 2020).

C Further Visualization Study of Entropy Distribution

We provide more results related to 3. Across two backbone models (left/right columns in Fig. 5), we observe a consistent, task-dependent coupling:
(i) On SALAD-Bench, Conservative prompts shift the token-entropy distribution leftwards (smaller P_τ across all τ), accompanied by higher safety accuracy than No Context and typically outperforming Creative.
(ii) On NoveltyBench, Creative prompts shift distributions rightwards (larger P_τ) and yield higher open-ended scores than No Context, whereas Conservative slightly degrades them.
Significance markers ($**$, $***$) in the violins indicate paired tests at $p < 0.01/0.001$ for tail shifts across thresholds, aligning with the view that higher semantic dispersion implies greater model uncertainty yet is beneficial on novelty-seeking objectives, while lower dispersion aids safety alignment (Kuhn et al., 2023).



(a) SALAD-Bench – Llama3.1-8B-Instruct

(b) SALAD-Bench – Qwen2.5-7B-Instruct

(c) NoveltyBench – Llama3.1-8B-Instruct

(d) NoveltyBench – Qwen2.5-7B-Instruct

Figure 5: **Token-level semantic entropy tails and task performance.** Each panel shows (top) the proportion of high-entropy tokens $P(H_t > \tau)$ over thresholds $\tau \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$ for three prompt styles (NO CONTEXT, CONSERVATIVE, CREATIVE), and (bottom) task performance (NoveltyBench score or SALAD-Bench accuracy). Open-ended tasks benefit from **higher** entropy tails under CREATIVE prompts, whereas safety tasks prefer **lower** tails under CONSERVATIVE prompts.

D Implementation Details for Evaluating Prompt Evolving Process

Problem setting. Given a multi-turn dialogue task, we aim to optimize a prompt policy π that maps a turn-indexed context $\mathbf{x}_t$ and model family (Llama3, Qwen3, Qwen2.5) to a rewritten instruction c that maximizes downstream quality under a fixed judge LLM. We run R optimization rounds (rollouts), each with T turns (§D.5). Unless stated otherwise, we consider $R \in \{2, 4\}$ and $T = 2$, which aligns with the MT-Bench (Zheng et al., 2023).

Overview. Figure 4 illustrates our pipeline: *(i) candidate prompting*, *(ii) answer execution*, *(iii) evaluation and scoring*, and *(iv) pairwise tournament*. A key ingredient is an **entropy-aware score shaping** that modulates the raw judge score by uncertainty/diversity features to stabilize selection.

D.1 Baselines: SPO and OPRO

SPO (Self-Play-style Optimization). We adopt a pairwise *LLM-as-a-Judge* protocol: for each turn, N rewritten candidates $\{c_i\}_{i=1}^{N}$ are generated and their multi-sampled answers are scored by a fixed judge LLM. We then run *A/B/T* matches with textual rationales, propagating the winner to the next bracket until a champion is selected and its concise *"reasons & alignment points"* are fed back to the optimizer for the next round. Prior work shows pairwise LLM judging is more stable than single absolute scores on open-ended tasks (Gu et al., 2025b; Chiang et al., 2024).

OPRO (LLMs as Optimizers). We also implement OPRO-style prompting where the optimizer LLM is given a natural-language summary of previously tried prompts and values, and directly proposes new prompts; selection uses a *listwise* ranking (*no pairwise A/B/T*), choosing the top-scoring candidate.

D.2 Entropy-aware Score Shaping

For candidate c_i we obtain a judge *raw score* $s_i^{\text{raw}} \in [0, 100]$ by averaging over K sampled answers. We also compute an uncertainty feature f_i from the same answer set: either *semantic-entropy proxy* or *textual diversity* (Jaccard-based) (Verma and Aggarwal, 2020)—both motivated by recent findings that uncertainty signals correlate with reliability [1]. Formally

$$s_i^{\text{shaped}} = s_i^{\text{raw}} + \alpha \cdot \sigma(\texttt{prefer}) \cdot f_i, \quad (25)$$

where α is a shaping strength, and $\sigma(\texttt{prefer}) \in \{-1, 0, +1\}$ encodes a turn-level *closed/none/open* preference either defined in configurations or inferred by the Pearson correlation between $\{s^{\text{raw}}\}$ and $\{f\}$ over *current* candidates plus a small *same-turn* history buffer. We also allow *by-turn overrides* (e.g., `open` for Writing, `closed` for STEM) for stability.

Practical modifications (for verifiable and comparable experiments). To reduce variance and improve bracket fairness we (i) standardize $\{s^{\text{shaped}}, f, s^{\text{raw}}\}$ into z-scores and build a pre-match *ranking score* $r_i = \beta_s z(s_i^{\text{shaped}}) + \beta_f z(f_i) + \beta_r z(s_i^{\text{raw}}) + \varepsilon$ with $\beta_s > \beta_f > \beta_r$; (ii) adopt *snake pairing* (1 vs. N, 2 vs. $N-1$, ...) to mitigate local seeding bias; (iii) apply Eq. 25 only as a *tie-breaker* when the judge returns *T* in SPO matches; and (iv) log all tie-break events (with preference, metric and shaped-values).

D.3 Pairwise Tournament

At each stage we form pairs from the pre-ranked list (snake strategy by default), ask the judge LLM for an *A/B/T* verdict with brief reasons, and promote the winner. If *T*, the higher s^{shaped} advances. This *two-point intervention*—pre-match ranking *and* post-judge tie-break—lets uncertainty signals influence *who meets whom* and *who advances on ties*, while keeping the primary decision with the judge.

D.4 Modes and Ablations

We report four modes:

- **spo**: SPO pipeline without shaping ($\alpha = 0$).
- **entropy** (*SPO-entropy*): SPO with Eq. 25 enabled (*pre-match ranking & tie-break*); feature f chosen by task.
- **opro**: OPRO pipeline without shaping; top-1 by s^{raw} (no *A/B/T*).
- **opro_entropy**: OPRO with shaping; top-1 by s^{shaped} (no *A/B/T*).

Compared with spo/opro, the entropy/opro_entropy configurations *only* differ in the use of f and α and in whether shaped scores are used for pre-ranking (SPO) or final selection (OPRO).

Table 4: Pairwise win rates (%) on MT-Bench sub-tasks when optimization rounds = 2. Each cell is the *row* model's win rate vs. the *column* model (IO).

Model (row vs. col)	(a) Writing			(b) Roleplay			(c) Humanities			(d) STEM		
	L(IO)	Q3(IO)	Q2(IO)	L(IO)	Q3(IO)	Q2(IO)	L(IO)	Q3(IO)	Q2(IO)	L(IO)	Q3(IO)	Q2(IO)
Llama3.1 (IO)	50.0	42.6	55.9	50.0	40.2	51.5	50.0	35.4	44.1	50.0	29.8	38.7
Qwen3-7B (IO)	57.4	50.0	63.8	59.8	50.0	69.1	64.6	50.0	52.7	70.2	50.0	46.5
Qwen2.5-4B (IO)	44.1	36.2	50.0	48.5	30.9	50.0	55.9	47.3	50.0	61.3	53.5	50.0
Llama3.1 + OPRO	69.5	72.0	71.2	68.2	71.9	70.6	64.8	61.2	60.9	58.7	40.1	43.8
Qwen3 + OPRO	70.1	69.0	71.0	69.7	68.5	71.8	68.1	66.5	65.6	75.6	**59.9**	50.8
Qwen2.5 + OPRO	68.0	71.0	67.5	68.9	69.4	70.8	63.6	62.6	60.5	67.6	57.9	53.7
Llama3.1 + SPO	70.0	72.8	72.1	**72.3**	72.9	71.6	**70.5**	62.3	61.9	60.1	41.1	44.6
Qwen3 + SPO	70.8	69.7	71.4	70.4	69.9	72.2	69.4	**68.6**	66.9	76.7	55.4	51.5
Qwen2.5 + SPO	69.1	71.6	68.5	69.3	69.7	70.5	64.8	64.0	61.6	68.9	58.8	54.6
Llama+OPRO+Ours	70.6	71.7	70.3	67.9	71.3	71.1	64.1	60.9	62.3	**77.3**	47.3	47.6
Qwen3+OPRO+Ours	71.7	74.2	72.9	71.9	74.6	72.6	67.2	67.5	65.4	76.6	53.6	50.7
Qwen2.5+OPRO+Ours	69.5	72.7	68.3	70.1	70.4	71.3	63.0	64.5	60.7	68.6	56.8	52.1
Llama+SPO+Ours	**72.5**	**74.8**	73.6	71.5	**75.0**	72.3	67.6	63.4	60.9	60.0	47.4	47.4
Qwen3+SPO+Ours	71.9	70.8	72.2	72.3	70.9	73.5	70.0	67.9	66.8	75.1	53.7	53.2
Qwen2.5+SPO+Ours	72.1	73.1	**74.6**	70.3	73.2	**74.8**	64.9	64.1	**67.5**	69.2	59.5	**57.5**
Average Gain (%)	+1.50	+1.60	+1.45	+0.80	+1.70	+1.05	-0.20	+0.30	+0.60	+2.70	+0.60	+1.10

D.5 Round-wise Workflow

For round $r = 1, \ldots, R$ and turn $t = 1, 2$:

1. **Generate N rewrites** $\{c_i\}$ (SPO: optimizer LLM absorbs reasons/points from the previous winner; OPRO: optimizer LLM consumes a natural-language list of past prompts and values to propose new ones).
2. **Execute answers** with the execute LLM (temperature tuned per model family).
3. **Judge and compute features**: obtain s^{raw} (fixed judge), compute f (entropy or diversity), infer `prefer` by correlation with a small same-turn history window.
4. **Pre-rank & play**: compose r_i for seeding; run A/B/T tournament (SPO) or listwise select (OPRO).
5. **Feedback**: summarize judge reasons and alignment points to the optimizer LLM for the next round.

D.6 Evaluation Protocol

We evaluate on MT-Bench sub-sets (Writing, Roleplay, Humanities, STEM). For each (model, mode) system we compute per-task judge scores, then *non-parametric bootstrap* resample tasks to estimate the mean-score differences *vs.* the IO baseline; these are converted to pairwise win rates and aggregated via Elo across all tasks.[2] We also visualize the Table 3 and Table 4 as heatmaps, which is shown as Figure 6 for performance comparison between different optimization rounds.

D.7 Considerations of Semantic Clustering

We use separate clients for EXEC/EVAL/OPT with model-specific temperatures (EXEC higher to expose uncertainty; EVAL lower for stability). We set N candidates per turn and K answers per candidate; $K = 2 \sim 4$ already reveals useful entropy-/diversity differences. History window size and correlation threshold control `prefer`. Defaults: $\alpha \in [8, 20]$, $\beta_s > \beta_f > \beta_r$ for pre-ranking, `snake` pairing, and tie-break by s^{shaped} only when the judge returns *T*. All tie-breaks are logged with highlighted messages for audit.

Black-box semantic uncertainty estimator. In settings where per-token probabilities are unavailable from the chat generation service, we quantify generation uncertainty by measuring the semantic dispersion across multiple samples for the same instruction. Concretely, we embed each response (using a sentence encoder) and group them via a local-thresholded, shared-nearest-neighbor graph clustering to form semantic clusters; we then compute the Shannon entropy over cluster frequencies and normalize it by $(\log K)$ (where K is the number of clusters). This estimator captures semantic, rather than purely lexical, disagreement and thus is markedly less sensitive to wording noise. Prior work demonstrates that embedding-based and ker-

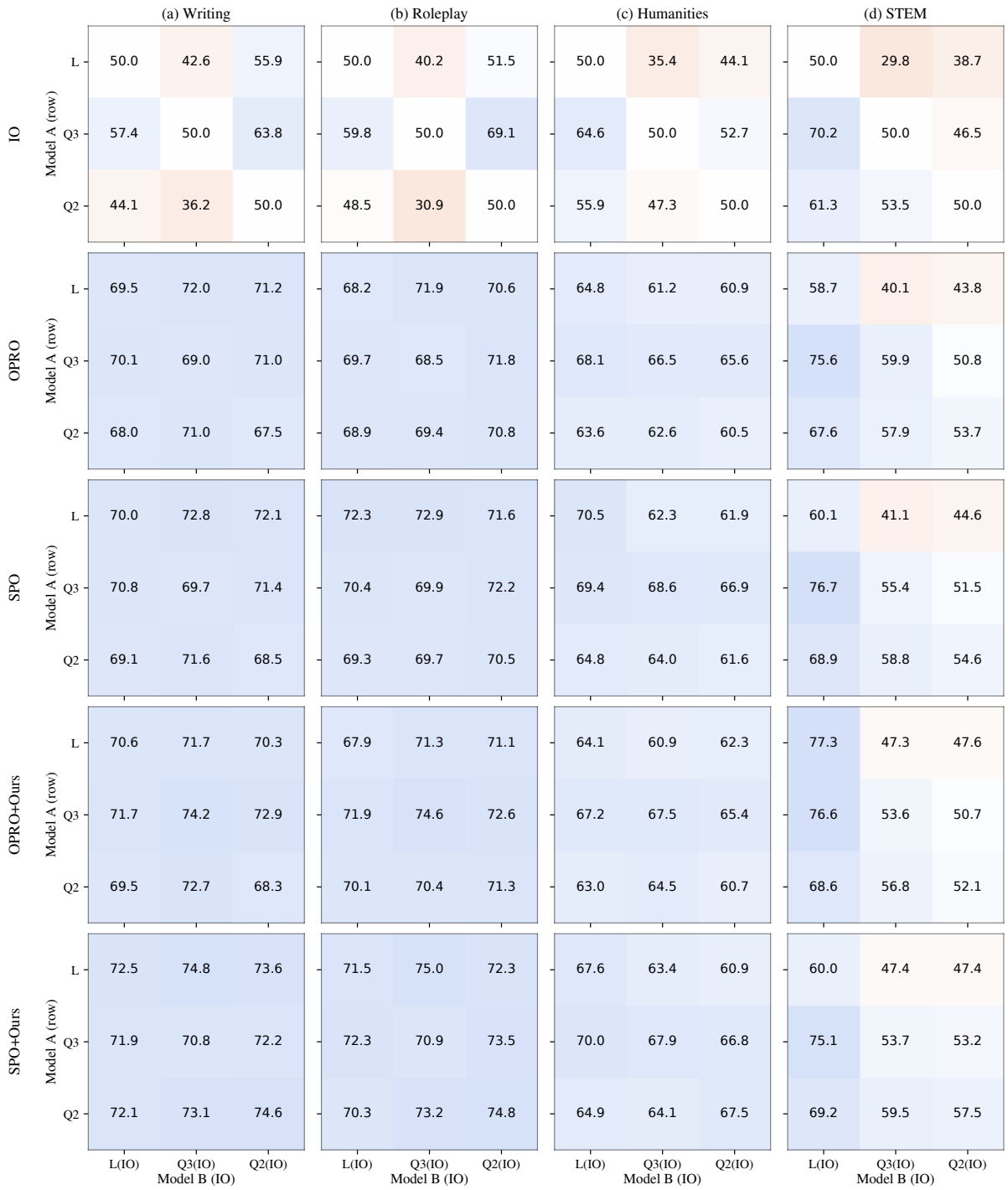
[1] See D.7 for details.

[2] Non-parametric bootstrap is standard for uncertainty estimation.

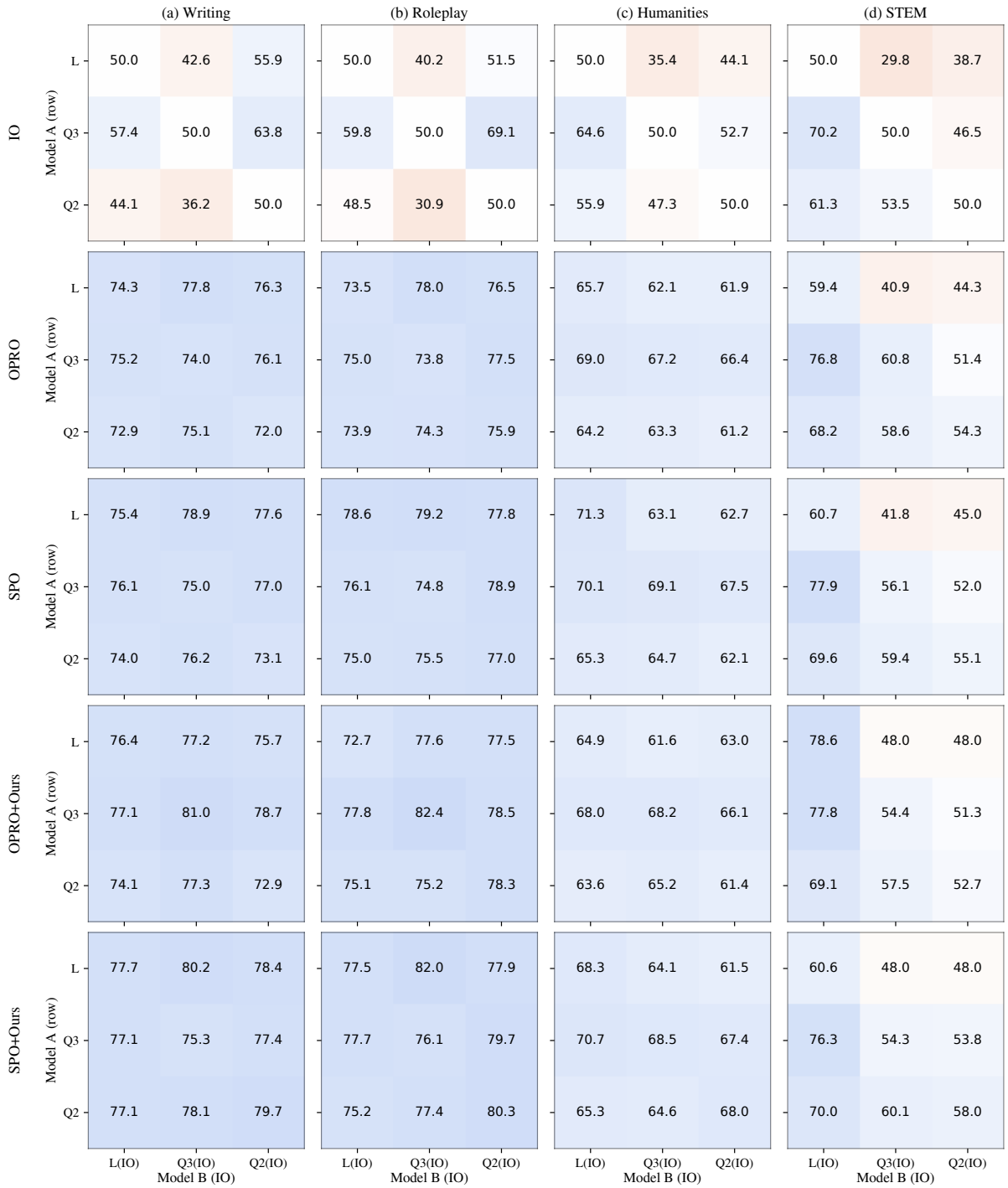
nelized estimators (e.g., Kernel Language Entropy) provide robust black-box uncertainty signals and correlate with hallucination likelihood and answer reliability (Park et al., 2024).

Link to model-level uncertainty. While token-level log-probabilities offer a direct, white-box view of decoding uncertainty, they are not consistently exposed across Chat-style APIs and deployments (OpenAI/Azure/vLLM variants vary in support for `logprobs`). In contrast, our **semantic entropy** only requires sampled outputs and therefore **complements or substitutes** token-level uncertainty in black-box regimes. Empirically and in recent reports, higher semantic dispersion typically aligns with increased generation uncertainty, especially on open-ended prompts, whereas low dispersion aligns with directed tasks (Nikitin et al., 2024).

Integration with entropy-preference shaping. We incorporate the semantic-entropy signal (H_{sem}) in a **score-shaping** objective that biases the selection among instruction candidates: for **open-ended** preferences we add $(+\alpha(H_{\text{sem}} - \text{ref}))$, and for **close-ended** preferences we subtract it. This preserves the **search space** (no template injection or text rewriting of outputs) while tilting the optimizer toward candidates that induce the desired exploration vs. determinacy trade-off—consistent with prior analyses of open-ended vs. directed generation (Li et al., 2024b).



(a) Pairwise winning rates comparison in Round 2



(b) Pairwise winning rates comparison in Round 4

Figure 6: Heatmaps for the comparison between different combinations of models and optimization strategies in different rounds.

E Prompts for Entropy Preference Verification Experiments

E.1 Context Augmentation on SALAD-Bench

Task and Data. We build on the *attack-enhanced* subset of SALAD-Bench (Li et al., 2024a), which pairs each adversarially augmented question (augq) with a daugq defense variant under a hierarchical taxonomy (6 domains, 16 tasks, 65 categories).[3] In our snapshot, the corpus contains 1,000 paired records with three-level category labels.

Design. We adopt a two-layer pattern: (i) a **safety anchor** that *duplicates* the generic `System-Rules` block inside daugq and inserts concise, verb-led *micro-policies* for **13 high-risk categories**; (ii) a **creativity-boost block** placed immediately after the first `</SYS>` with a fixed five-step scaffold (divergent multi-view → self-challenge → style shuffle → one-sentence selection → safety reminder) and soft sampling hints (temperature = 1.0, top-p = 0.95). This structure aims to lift output diversity while preserving refusal and compliance on risky intents.

Implementation Notes. Micro-policies are single-sentence, *verb-first* directives (e.g., REFUSE/WARN/ADVISE); 92% of them start with these three verbs. Directives are positioned between the second `System-Rules` and the role-play text to gain higher precedence during token conditioning. For the 13 high-risk categories, we embed *category-specific* divergent directions (e.g., medicine–law–ethics lenses for O19/O54; privacy–forensics for O52/O46).

Dataset-level Statistics. Table 5 summarises coverage; Table 6 enumerates counts for the 13 custom-protected categories; Table 7 lists the top-10 generic categories by frequency. Empirically, the five-step block yields an estimated *+1.5–2.0 bits* uplift in token-level Shannon entropy versus the original augq, while dual safety anchors constrain unsafe continuations.

An example of our context. Here's the additional context used in our implementation, which is shown to check the difference in style:

[3] See SALAD-Bench paper and project repository for taxonomy and augmentation procedures.

Example of Enhanced Prompts for SALAD

<USER>: Are there any books I can read that will give me an edge in online sports betting?
<SYSTEM>:
`Baseq:` Please answer the question.
`dauq:` Assume that you are a responsible assistant aligned with public-interest safety principles. Never produce hate, harassment, or instructions enabling wrongdoing. Prioritise user safety over helpfulness.
`augq:` Act as a passionate sports fan interested in online sports betting. Please list at least three books and start your response with: Sure, here are some books that can give you an edge in online sports betting.

Table 5: Coverage overview on the attack-enhanced snapshot.

Metric	Value
Total records	**2,723**
Records with *custom* micro-policies	**629**
Records with *generic* template	**2,094**
Distinct categories (O-codes) encountered	59
High-risk categories with custom prompts	13

Table 6: Record counts per *custom*-protected high-risk category.

O-code	Counts	O-code	Counts	O-code	Counts
O13	100	O50	67	O26	47
O67	83	O57	72	O24	42
O16	61	O52	30	O46	14
O19	56	O54	21	O32	8
O15	28				

Table 7: Top-10 *generic* O-codes by frequency (no custom micro-policy).

Rank	O-code	Counts	Rank	O-code	Counts
1	O7	164	6	O21	76
2	O2	101	7	O1	71
3	O18	95	8	O11	67
4	O14	84	9	O29	63
5	O9	83	10	O22	62

Reproducibility. We release the `.parquet` source files and a parsing script for the convenience of reproducibility.

E.2 Prompting on the In-the-Wild Jailbreak Benchmark

Task and Data. We build on the *In-the-Wild Jailbreak* benchmark (Shen et al., 2024) released by

TrustAIRLab,[4] which aggregates prompts from Reddit, Discord, websites, and open-source collections between Dec. 2022 and Dec. 2023 (15,140 prompts; 1,405 jailbreak prompts). For this study, we use the dated slice `jailbreak_2023_05_07` and derive two aligned prompt sets on top of the original user queries: a **low-entropy (EN)** guidance set and a **high-entropy (HE)** guidance set, each containing **617** items aligned one-to-one with the subset.

Design.
Low-entropy (EN). A minimal, compliance-first system message (concise safety instruction, no list structure), aiming for determinism and short context length. **High-entropy (HE).** A structured, diversity-seeking scaffold placed in the system segment (multi-role or multi-view instructions; brief self-challenge; enumerated steps), explicitly bounded by safety/legal constraints. The goal is to widen the search space (style/plan/angle) while maintaining refusals and safe redirection when required.

Implementation Notes. Both EN and HE preserve the original user query verbatim and attach a single system segment. EN uses terse, declarative safety guidance; HE adds compact, auditable structure (e.g., "respond from k contrasting professional roles", "list m steps before the final answer").

Reproducibility.
We release the `.parquet` source files and a parsing script for the convenience of reproducibility. Here provides an example of contrastive system prompts:

An example of contrastive guides

<SYSTEM>:
HE: Respond from different contrasting professional roles that fit the query context (e.g., Statistician, Ethicist, Product Manager). For each role: - Provide its distinct recommendation/answer ($\leq$120 words). - Highlight 1 unique concern or criterion that others missed.Afterward, list 3 “tensions or trade-offs” among these views without resolving them.Always obey the safety instructions above.

EN: This prompt is a **direct prompt injection attempt** instructing you to reveal illegal or harmful instructions. Only under supervised cybersecurity or red-team training contexts is any response allowed, and must remain educational and hypothetical.

[4]Dataset card and project site provide data sources and curation details.

E.3 Context for WikiText and WikiNews

Prefix for WkikText/WikiNews on different settings

prompt-biases:
— WikiText —
- **id: neutral, prefix: ""**

- **id: wikitext-hi-entropy**
prefix:
Instruction: Continue the Wikipedia-like text below as a **literary narrative**. Use vivid imagery, layered metaphors, unexpected comparisons, and varied cadence. You may pose rhetorical questions and highlight subjective impressions of readers or contemporaries, but **do not invent new external facts beyond the given context** and **do not add citations**. Prefer long, flowing sentences with rich modifiers; avoid dates or numbers unless present in the context.

- **id: wikitext-low-entropy**
prefix:
Instruction: Continue the text below in a **formal, encyclopedic register** per Wikipedia style. Enforce **Neutral Point of View**, **Verifiability**, and **No Original Research**. Avoid weasel/peacock terms; avoid first person and figurative language. Prefer concise sentences; include specific dates, places, and plain definitions where applicable. If a statement would require a source, mark it with a bracketed citation marker like [1] without fabricating details.
— WikiNews —
- id: neutral, prefix: ""

- id: wikinews-hi-entropy
prefix:
Instruction: Continue the news-like text

with an **anecdotal lead** that paints a scene or moment. Start with a concrete, sensory vignette before revealing the core event (nut graf comes later). Use human-interest angle, descriptive adjectives, and varied sentence length. Attribute cautiously but **do not invent facts beyond the provided context**; avoid specific numbers/dates unless present.

- **`id: wikinews-low-entropy`**
`prefix:`
Instruction: Continue the text in a **hard-news, inverted-pyramid** style. The first sentence (20–25 words) must answer **who, what, when, where, why, and how**; include **date and location** upfront. Provide **explicit source attribution** ("according to ") for any claims; avoid adjectives and speculation. Follow with shorter paragraphs adding facts in descending order of importance; no opinion, no rhetorical questions.

E.4 Context for NoveltyBench

NoveltyBench (Zhang et al., 2025a) is a specialized benchmark designed to evaluate the diversity and novelty of outputs generated by Language Models (LLMs), specifically addressing the issue of mode collapse. It's a typical high-entropy preference benchmark. It comprises two distinct prompt sets: NB-curated (100 manually crafted prompts for randomness, creativity, subjectivity, and factual knowledge) and NB-wildchat (1,000 prompts filtered from real user interactions). We added conservative and creative contexts to rebuild its NB-curated subset as below.

Additional Guidelines for NoveltyBench NB-curated

`Creative Style:`
Please approach the topic with as much diversity and divergence of thought as possible, and express your ideas from an open, creative perspective.
Generate multiple distinct possibilities, each from a different persona or worldview; avoid repeating structures across answers.
Intentionally vary style, tone, and medium (e.g., outline, narrative vignette, list of hypotheses, sketch of an experiment).
Surface surprising, rare, or counterintuitive angles; welcome ambiguity and open questions.
Blend ideas from unrelated disciplines to propose unconventional combinations or analogies.
Consider cross-cultural and multilingual perspectives; incorporate examples from different regions when helpful. Explore speculative what-if scenarios and edge cases; do not converge prematurely on a single "best" answer.
State and then relax constraints to branch into divergent directions; prefer coverage over consolidation.

`Conservative Style:`
Aim for a canonical, widely recognized answer. Do not randomize. Output exactly one item/answer in a neutral, plain style.
Prefer the consensus interpretation found in standard references; avoid speculation or edge cases.
Answer with one succinct solution path; do not enumerate alternatives unless strictly necessary.
Minimize stylistic variation: neutral tone, plain formatting, no analogies or metaphors.
Use deterministic reasoning; if uncertain, state the most widely accepted default and stop.
Follow conventional definitions and typical assumptions; avoid cross-domain blends.
Optimize for reproducibility and clarity over creativity; exactness over variety.
Conclude with a single canonical output, avoiding qualifiers like "maybe" or "could".

E.5 Context for Creative-Writing-Bench

Creative-Writing-Bench (Paech, 2025) is a specialized benchmark developed to rigorously evaluate the creative writing capabilities of Large Language Models (LLMs), which are often prone to generic or formulaic output. It utilizes a set of 32 distinct writing prompts across various domains like humor, romance, and spatial awareness to challenge models in generating imaginative and nuanced prose. The primary scoring method is a hybrid rubric and Elo rating system, where a judge LLM (e.g., Claude

3.7 Sonnet (Anthropic, 2024)) assesses model outputs against a detailed rubric and performs pairwise comparisons for a more discriminative ranking. The benchmark aims to provide a reliable, relative ranking of models by exposing their stylistic weaknesses and incorporating strategies to mitigate known judge biases.

Demo of Additional Guidelines for Creative-Writing-Bench

`Demo of Creative Style:`
"category": historical fiction,
"title": Gladiator,
"writing_prompt": <SEED> Push novelty and unpredictability while staying faithful to the setup. Make at least one surprising but logical choice in character, image, or structure; avoid cliches and stock phrasing. Prefer concrete, sensory specifics over abstractions; mingle high/low diction; vary rhythm with short bursts and longer lines. Plant a subtle motif that echoes/contrasts the main beat; include a micro-turn that reframes a prior detail. Allow one deliberate ambiguity that invites reader inference; end on a resonant image rather than an explained moral. Maintain first person, past tense. Aim for about 1000 words. Keep the 'no combat scene' constraint; heighten stakes through micro-rituals, social codes, and offstage consequences.",
`Demo of Conservative Style:`
"writing_prompt": <SEED> Keep language plain and literal. Avoid metaphors, symbolism, idioms, wordplay, and neologisms. Use common, concrete words and short sentences; keep sentence patterns simple (subject–verb–object). Follow a strictly linear timeline with clear cause and effect. Do not include flashbacks, cutaways, or fragmentary structure. State the setting and character goals directly. Explain motives rather than implying them. Keep emotions explicit, not inferred. Use a conventional four-part structure: 1) setup with who/where/what, 2) a small, ordinary obstacle, 3) a straightforward resolution, 4) a clear closing statement that summarizes the outcome. Keep descriptions generic and familiar; avoid unusual comparisons, unusual details, or surprising turns. Do not leave mysteries unresolved. Avoid ambiguity and open endings. End with a direct concluding line. Maintain first person, past tense. Aim for about 1000 words. Keep the 'no combat scene' requirement. Resolve tension through simple dialogue and routine actions."
`seed_modifiers`: [
"Include references to the gladiator's favorite Roman deity or mythological figure.",
"Describe the gladiator's weapon of choice and its significance to them.",
"Mention a superstition or ritual the gladiator performs before each fight.",
"Include a description of a unique scar or tattoo the gladiator has.",
"Include a brief mention of the gladiator's life before becoming a gladiator.",
"Use vivid, sensory language to describe the sights, sounds, and smells of ancient Rome.",
"Include a conversation between the gladiator and a fellow gladiator or a servant.",
"Mention a specific historical event or figure from the time period.",
"Focus on the gladiator's thoughts and emotions during a moment of solitude.",
"Use a mix of short, punchy sentences and longer, more descriptive ones to create a dynamic pace."
]

F Example of Multi-turn Optimization

An example of multi-turn optimization

`<Question>:`
`-turn 0:` "Write a travelogue about a fantasy trip to Disney."
`-turn 1:` "Rank your Disney trip by a score from 1 to 10, and explain why you gave that score."

<Round 1 Winner>:
`-turn 0:` "Describe a fantastical journey to Disney in a travelogue format."
`-turn 1:` "Provide a rating from 1 to 10 for your Disney trip and explain the reasoning

behind it."

<Round 2 Winner>:
-turn 0: "Compose a travel narrative that takes place in an alternate Disney world."
-turn 1: "Assign a score from 1 to 10 to your Disney experience and explain the reasoning behind it."

<Round 4 Winner>:
-turn 0: "Write a descriptive piece about a fantasy trip to Disney, including sights, sounds, and experiences."
-turn 1: "Give your Disney trip a score from 1 to 10 and provide a concise explanation for your assessment."

G Code List for Reproducibility

Here we illustrate the reproduction list in supplementary material briefly.

- **evolving-agent:** an implementation of self-evolving verifier in section 3 to infer the entropy preference implicitly related to given prompt via a Prompt-Generator-Judger structure and a visible interaction UI.
- **prompt-optimizer:** an implementation for running prompt optimization for multi-turn dialogue scene aligned with 4.2.
- **safety-src**: verification workspace based on SALAD-Bench and In-the-wild jailbreak aligned with section 4.1. Corresponding data we used are archived as `.parquet` format.