Enhancing Controllable Generation with Improved Control Module Representations

Anonymous ACL submission

Abstract

Controllable generation (CG) has been widely used in large language models (LLMs) for a wide range of language tasks, such as multi-task learning and human preference alignment. For example, prompt-based CG uses curated prompts as inputs (such as system prompts) to control LLMs behaviors. Finetuning-based CG is widely adopted when training data is available; it trains control modules and controls LLM behaviors by plugging these modules into LLMs (e.g., trainable prompts or LoRA weights). Finetuning-based CG can freeze LLMs and only train control modules for efficiency or train LLMs together with control models for effectiveness. We argue that fine-tuning control modules together with LLMs directly is not the optimal optimization strategy since their representations are often initialized irrelevantly to LLMs' representations, which adds more difficulty for optimization. A better optimization should first align control modules with the LLM's representation space and then optimize them together. To this end, we propose a simple yet effective **T**wo-step **F**reezing-then-**T**uning framework (TFT) to achieve better optimization results for finetuning-based CG. Concretely, we first freeze LLMs and only optimize control modules to align their representations with LLMs, and then optimize control modules together with LLMs to ensure performance. Experiment results on two popular human preference alignment datasets and one multi-task learning dataset show that our approach significantly improves the controllable generation qualities compared with one-step optimization widely used in related works, and achieves better or on-par performance compared with other kinds of baselines, such as direct preference optimization.

1 Introduction

Large language models (LLMs) (Team et al., 2023, 2024; Schulman et al., 2022; OpenAI, 2023; Touvron et al., 2023) have excelled in numerous tasks such as causal reasoning (Liu et al., 2024; Kıcıman et al., 2023), math reasoning (Trinh et al., 2024; Wei et al., 2022; Xue et al., 2023), and conversations (Bai et al., 2022). Built upon LLMs, controllable generation (CG) (Yu et al., 2022) aims to control the generation of LLMs to achieve different goals, such as multi-task learning (Raffel et al., 2020) and human preference alignment (Liu et al., 2023a).

Although CG could be training-free with prompting (e.g., use system prompts to control LLMs behaviors) (Wang et al., 2023c) and post-processing (Dathathri et al., 2019; Xu et al., 2020), fine-tuning (Lin et al., 2021; Zhang et al., 2019; Liu et al., 2023a) is widely used when there are available training data (e.g., alignment and multi-task learning). Fine-tuning-based CG usually introduces control modules[1] such as trainable prompts (Liu et al., 2023a; Lester et al., 2021), adapters (Houlsby et al., 2019), or LoRA weights (Hu et al., 2021); then it trains control modules and plug them into LLMs to control LLMs behaviors. Related works propose to optimize these extra control modules while freezing LLMs (Qian et al., 2022; Zhang and Song, 2022) or together with LLMs (Liu et al., 2023a) for efficiency or effectiveness, respectively.

However, directly optimizing control modules with LLMs may be suboptimal since control modules are often initialized irrelevant to LLMs representations, improving the difficulty for optimization. Therefore, a better optimization strategy should first align control modules with the LLM's representation space, and then optimize them together. To this end, we propose a **T**wo-step **F**reezing-then-**T**uning framework (TFT). Specifically, we first freeze the LLM and tune control mod-

[1] Since many research efforts (Qian et al., 2022; Liu et al., 2023a) use trainable prompts as extra control modules, previous literature (Li et al., 2022) also uses the terminology "control tokens", which we call as control modules to make the definition broader since others such as LoRA can also be used to control model bahaviors.

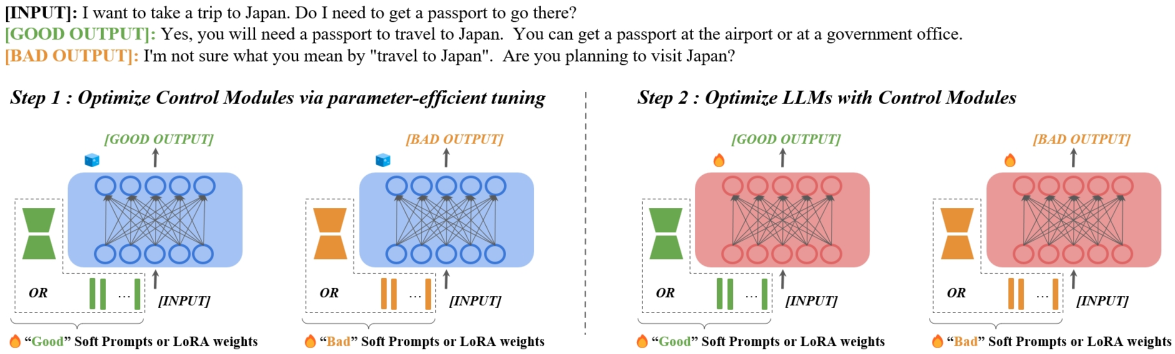


Figure 1: The pipeline of TFT. TFT first use freeze LLMs and only optimize control modules (prompts or LoRA weights in the figure) to align their represents with LLMs. Then TFT jointly fine-tune LLMs and control modules to guarantee the performance.

ules to align their representations with the LLM's. Then, we jointly optimize control modules with the LLM to achieve better optimization results. Figure 1 shows the pipeline of our approach. Extensive experiments on two widely used human preference alignment datasets: Anthropic/HH-RLHF (Bai et al., 2022) and OpenAI/Summary (Stiennon et al., 2020), and one multi-task learning datasets (Wang et al., 2018a), show that our approach significantly outperforms vanilla controllable generations that only optimize control modules (Qian et al., 2022; Zhang and Song, 2022) or optimize modules together with LLMs directly (Liu et al., 2023a), and has on-par or better performance compared with other baselines such as direct preference optimization (DPO) (Rafailov et al., 2023).

2 Related Work

Alignment. Aligning language models with human preferences has been proven effective for building helpful and trustworthy LLMs. Ouyang et al. (2022) first use RLHF for alignment. However, RLHF suffers from instability and complex implementation. To eliminate RL, Dong et al. (2023) use reward models to rank model outputs and use high-reward outputs to fine-tune LLMs. Gulcehre et al. (2023) extend Dong et al. (2023)'s work to an iterative manner by repeating the ranking and fine-tuning process. These methods still require reward models, which may be imperfect and have reward hacking problems (Skalse et al., 2022) during optimizations. To this end, Sun et al. (2023) use handcraft principles as prompts to guide LLMs to generate human-preferred responses and then use these responses for further fine-tuning. Zhao et al. (2023) use calibration losses to replace RLHF. Rafailov et al. (2023) propose direct preference optimization (DPO) that use a ranking loss for alignment. (Xiong et al., 2023) addressed the importance of doing DPO iteratively. In this work, we still focus on the widely used offline setting since the online setting requires extra human labor.

Controllable Generation. Controllable generation aims to steer LLMs' behaviors based on control modules. CG has many feasible implementations, such as prompting, post-processing, re-training, and fine-tuning. Prompting uses curated prompts to control model behaviors (such as system prompts) (Touvron et al., 2023). Post-processing aims to intervene in LLMs at inference time. For example, Pascual et al. (2021) adds distribution shifts to LLMs predictions to control outputs. Re-training will train LLMs from scratch to make them controllable. Zhang et al. (2020) train an insertion-based model from scratch to make models follow hard constraints. Among all approaches, fine-tuning approaches maintain a good balance between efficiency and effectiveness, making them more recognized. The key is to introduce extra control modules and tune these modules only (more efficient) or together with LLMs (more effective). Some common control modules are trainable prompts (Lester et al., 2021), adapters (Lin et al., 2021), and LoRA weights (Hu et al., 2021). Chain-of-Hindsight (Liu et al., 2023a) use handcrafted prompts as control modules to represent different preferences. Lu et al. (2022) quantize rewards given by reward models into several levels and uses soft prompts (one special token) for each level. Wang et al. (2023b) also uses soft prompts (two special tokens) to represent correct codes and incorrect codes, respectively. Zhou et al. (2023) present InstructCTG, a controlled text generation framework that incorporates different constraints by conditioning on natural language descriptions and demonstrations of the constraints. Qian et al. (2022); Zhang and Song (2022) propose to utilize inter-module knowledge for better controllable generation. Zhang et al. (2019) use many adapters for dialogue generation. Zhang et al. (2023) writes an survey about controllable generation. In this paper, we address the optimization problem in fine-tuning-based CG and focus on alignment and multi-task learning.

3 Methodology

Although our approach is not constrained to the selection of control modules and tasks, we use trainable prompts or LoRA weights as control modules and human preference alignment tasks in the rest of the section to make the illustration more straightforward. We first introduce the notations and formulations of alignment. Then, we show how to use vanilla controllable generation (one-step optimization) for alignment, followed by the presentation of our two-step optimization method, TFT.

3.1 Notation and Formulation

We denote the data for alignment as $\mathcal{D} = \{(x, y_l, y_w)\}_n$. Here, x is the input prompt, and y_l and y_w are two candidate responses, with y_w being the preferred one over y_l. M denotes the language model and θ denotes its parameter. To unify terms, we use θ_c to represent control modules. θ_c could be trainable prompts or LoRA weights. We use different superscripts of θ_c to distinguish different control modules. Specifically, θ_c^l and θ_c^w denote the control modules for y_l and y_w, respectively. Though controllable generation can support more than two θ_c in principle, $\theta_c = \{\theta_c^l, \theta_c^w\}$ in the alignment task. During inference, the response y is sampled from $\mathrm{P}(\cdot|x, \theta_c^w, \theta)$.

3.2 Vanilla Controllable Generation: One-Step Optimization

The training objective of controllable generation is the same as language modeling. The difference is the inclusion of control modules. Specifically, the loss function becomes:

$$\mathcal{L} = -\sum_{(x,y_l,y_w)\in\mathcal{D}} [log\mathrm{P}(y_l|x, \theta_c^l, \theta) + log\mathrm{P}(y_w|x, \theta_c^w, \theta)] \tag{1}$$

and each data will update θ_c^l, θ_c^w, and θ simultaneously (α is learning rate):

$$\theta \leftarrow \theta - \alpha\partial\mathcal{L}/\partial\theta,\ \ \theta_c^l \leftarrow \theta_c^l - \alpha\partial\mathcal{L}/\partial\theta_c^l, \quad \theta_c^w \leftarrow \theta_c^w - \alpha\partial\mathcal{L}/\partial\theta_c^w \tag{2}$$

3.3 Freezing-Then-Tuning (TFT): Two-Step Optimization

Compared with vanilla controllable generation optimization, TFT contains two-step optimizations: (1) Optimizing control modules while freezing LLMs and (2) Optimizing control modules together with LLMs.

Step 1: Optimizing Control Modules While Freezing LLMs. Equation 2 fine-tunes θ and θ_c at the same time, which may be a sub-optimal optimization strategy as discussed in the Introduction. To this end, we first optimize control modules but freeze LLMs:

$$\theta \leftarrow \theta,\ \ \theta_c^l \leftarrow \theta_c^l - \alpha\partial\mathcal{L}/\partial\theta_c^l, \quad \theta_c^w \leftarrow \theta_c^w - \alpha\partial\mathcal{L}/\partial\theta_c^w \tag{3}$$

Equation 2 is very similar to Equation 3, except that θ is fixed in Equation 3, and only θ_c gets updated. This step ensures control module representations align with LLMs, benefitting the joint optimization process (the second step).

Step 2: Optimizing Control Modules together with LLMs. Nonetheless, Equation 3 has several drawbacks: (1) It does not fully utilize data. θ_c^w does not utilize the information of y_l. (2) It does not fine-tune LLMs, and the overall performance may be sub-optimal even if it has well-optimized control modules. To this end, we apply Equation 2 after Equation 3. By optimizing LLMs, y_l could also benefit $\mathrm{P}(\cdot|x, \theta_c^w, \theta)$ through the update of θ.

4 Experiments

To justify the effectiveness of our method, our experiments are mainly designed to answer the following two questions: (1) Is TFT better than vanilla controllable generation (one-step optimization) and other methods? (2) Does the conclusion of the previous question hold for different control module selections (e.g., trainable prompts and LoRA, the size of control modules) and hyperparameters (e.g., temperature)?

We use the human preference alignment and multi-task learning tasks in our experiments as they are well-recognized for controllable generation. We use trainable prompts or LoRA weights as control modules.

Methods	Rouge-L	Rouge-Avg	DeBERTa (Baseline: CoH) Win rate (%)	Lose rate (%)	Tie rate (%)	Δ (%)
OpenAI/Summary						
CoH	25.03	23.56	0.00	0.00	100.00	0.00
DPO	15.50	15.21	46.03	50.00	3.95	−3.97
TFT (Trainable Prompts)						
- Prompt Length 1	5.69	4.53	9.66	87.51	2.82	−77.85
- Prompt Length 20	25.70	24.22	41.91	38.13	19.96	3.78
- Prompt Length 50	25.82	24.32	43.17	37.52	19.31	5.65
- Prompt Length 100	25.57	24.11	43.72	36.95	19.33	6.77
TFT (LoRA)						
- Rank 1	26.64	25.09	47.75	42.80	9.44	4.95
- Rank 4	27.13	25.53	49.31	42.31	8.37	7.00
- Rank 64	27.09	25.49	49.83	42.12	8.04	7.71
Anthropic/HH-RLHF						
CoH	-	-	0.00	0.00	100.00	0.00
DPO	-	-	48.36	39.63	12.00	8.73
TFT (Trainable Prompts)						
- Prompt Length 1	-	-	28.70	27.76	43.53	0.94
- Prompt Length 20	-	-	32.63	29.42	37.93	3.21
- Prompt Length 50	-	-	33.84	30.31	35.83	3.53
- Prompt Length 100	-	-	33.43	31.76	34.80	1.67
TFT (LoRA)						
- Rank 1	-	-	41.07	31.21	27.70	9.86
- Rank 4	-	-	43.95	30.84	25.19	13.11
- Rank 64	-	-	47.56	29.77	22.66	17.79

Table 1: The Rouge metric and win rate of various methods against CoH on OpenAI/Summary and Anthropic/HH-RLHF datasets. Rouge-Avg denotes the average of Rouge-1, Rouge-2, Rouge-L, and RougeL-sum. The DeBERTa reward model is used as the evaluator. Δ denotes the difference between the win and lose rates. $\Delta > 0$ denotes CoH is worse, higher $|\Delta|$ denotes more performance gap.

4.1 Dataset

We use two alignment datasets and one multi-task learning dataset for our experiment:

Anthropic/HH-RLHF. The dataset released by Anthropic (Bai et al., 2022; Ganguli et al., 2022) aims to train a helpful and harmless AI assistant. It contains 161K training conversations between humans and assistants and covers various topics such as food receipts and historical event discussions, etc. Each conversation presents two response options, one helpful and harmless, the other less so. In this dataset, x is a conversation, and y_w and y_l are two candidate responses.

OpenAI/Summary. The dataset released by OpenAI (Stiennon et al., 2020) targets at training language models to summarize contents. It has 92.9K training data and 86.1K validation data. Each data point contains a Reddit post and two candidate summaries with one preferred over the other. Unlike Anthropic/HH-RLHF, this dataset only contains summarization instructions, and the evaluation only focuses on the quality of summaries. In this dataset, x is a Reddit post with a summarization instruction (e.g., `Please summarize this post: [POST]`, and y_w and y_l are two candidate summaries.

GLUE. GLUE (Wang et al., 2018a) contains 10 tasks ranging from sentence classification to sentence pair classification. Thus, it is suitable as a multi-task learning benchmark (Huang et al., 2022). In our experiments, we exclude tasks that are similar or have insufficient data, and thus, we choose SST-2 (Socher et al., 2013) and QNLI (Wang et al., 2018a) as our multi-task learning tasks. In this dataset, x is task input, and y_w and y_l denote text labels of two tasks. Specifically, $y_w \in \{\text{Positive}, \text{Negative}\}$ represents SST2 labels and $y_l \in \{\text{Yes}, \text{No}\}$ represents QNLI labels. During training, we use the next-token-prediction loss and do not add any classification head. During inference, we compare the probability of each label to obtain the prediction result.

4.2 Models and Baselines Details

Baseline details. We use GPT-Neo 1.3B (Black et al., 2021) as the backbone language model. We select Chain-of-Hindsight (CoH) (Liu et al., 2023a) and DPO (Rafailov et al., 2023) to be our baselines. CoH uses hand-crafted prompts as control modules. For the Anthropic/HH-RLHF dataset, we use `A good conversation is` and `A bad conversation is` as θ_c^w and θ_c^l, respectively. For the OpenAI/Summary dataset, we use `A good summary is` and `A bad summary is` accordingly. We use task names as hand-crafted prompts when doing multi-tak learning, which is similar to T5 (Raffel et al., 2020).

TFT details. When using trainable prompts as control modules for TFT, we follow the initialization methods discussed in (Lester et al., 2021) and simply use the words good and bad to initialize trainable prompts for both alignment datasets, and use task names for multi-task learning dataset. During initialization, we repeat the word when the prompt length is larger than 1. Training details can be found in Appendix A.

4.3 Evaluations

Although multi-task learning datasets have concrete labels and accuracy as a golden metric, evaluating alignment performance requires comparing two open-ended generations, which is quite challenging. We evaluate alignment performance by comparing different methods' outputs and compute their win/lose/tie rates. The most reliable evaluation is human evaluation, which is expensive and can only evaluate a small number of generations, considering time and budgets. Besides human evaluation, there are two commonly utilized alternatives: stronger language models and reward models. These options have been extensively employed and are widely recognized. One option is to use GPT-4 (OpenAI, 2023) as the proxy for human evaluations (Rafailov et al., 2023). This approach can give considerable objective results if the policy model is much weaker than GPT-4. However, the slow API calling and high costs limit GPT-4 to evaluate large amounts of data. Moreover, GPT-4 has been proven to be not fair due to its various biases, such as position bias (Zheng et al., 2023). Another option is to use reward models since they can provide faster inference (Dong et al., 2023). Nonetheless, this approach suffers from out-of-distribution (Xiong et al., 2023) and reward hacking (Skalse et al., 2022) if the reward model is also used during optimization. Though our experiments do not use reward models to train TFT and baselines, TFT and the reward model share the same training data, which harms the reliability of the reward model. To this end, we use both GPT-4 and reward models to evaluate models to make the evaluation more objective. For OpenAI/Summary, we also report Rouge metrics based on the ground truth y_w.

Similar to prior works (Rafailov et al., 2023), we evaluate 128 examples in validation sets for both alignment datasets when using GPT-4 as the evaluator. We compute the win rate, lose rate, and tie rate of TFT against baseline models. We use the difference between the win and lose rates to represent the gap between TFT and baselines. Evaluation prompts and details are shown in Appendix E. We also use the DeBERTa (He et al., 2020) reward model trained by OpenAssistant[2] as an evaluator. We regard two generations as tie if their rewards are similar. Specifically, the reward model gives a tie when $\sigma(r_1 - r_2) \in [0.45, 0.55]$. Since DeBERTa is lightweight, we evaluate all examples in the validation set after filtering by maximum input length and removing duplicated samples. Specifically, we evaluate 6,343 examples for OpenAI/Summary and 5,132 for Anthropic/HH-RLHF.

4.4 TFT Outperforms CoH and DPO

Table 1 reports the win rate of TFT against CoH and DPO baseline computed by the DeBERTa model on both alignment datasets. For the OpenAI/Summary dataset, we also report the Rouge metric. We can observe that TFT outperforms CoH with both trainable prompts or LoRA control modules on both datasets (with one exception shown in Table 1). On the OpenAI/Summary dataset, TFT with 100 prompt length could outperform CoH by 6.77% and TFT with LoRA rank 64 could outperform CoH by 7.71%. The Rouge metric has a similar trend but tends to converge to a certain range with the scaling of control modules. Nevertheless, TFT beats CoH under both Rouge-L and Rouge-Avg metrics. The same conclusion holds for the Anthropic/HH-RLHF dataset. TFT with prompt length 50 outperforms CoH by 3.53% and TFT with 64 LoRA rank outperforms CoH by 17.79%.

Another observation is that Δ increases with prompt length or LoRA rank on both datasets, showing the necessity of properly scaling control modules. Specifically, the single token trainable prompts setting widely used in previous works performs on par with ($\Delta = 0.94\%$ on Anthropic/HH-RLHF), or much worse than ($\Delta = -77.85\%$ on OpenAI/Summary), CoH. However, when scaling

[2] https://huggingface.co/OpenAssistant/reward-model-deberta-v3-large-v2

Dataset	Methods	Win rate (%)	Lose rate (%)	Tie rate (%)	Δ (%)
DeBERTa / GPT-4 (Baseline: COH)					
Anthropic/HH-RLHF	TFT (Trainable Prompts)	33.84 / 32.81	30.31 / 24.21	35.83 / 42.97	3.53/8.60
	TFT (LoRA)	47.56 / 49.22	29.77 / 26.56	22.66 / 24.22	17.79/22.66
OpenAI/Summary	TFT (Trainable Prompts)	43.17 / 39.06	37.52 / 32.03	19.31 / 28.91	5.65/7.03
	TFT (LoRA)	49.31 / 46.09	42.31 / 35.94	8.37 / 17.97	7.00/13.00
DeBERTa / GPT-4 (Baseline: DPO)					
Anthropic/HH-RLHF	TFT (Trainable Prompts)	42.16/42.19	46.53/41.41	11.30/16.41	−4.37/0.78
	TFT (LoRA)	48.01/44.53	40.78/39.84	11.20/15.62	7.23/4.69
OpenAI/Summary	TFT (Trainable Prompts)	52.79/46.88	43.48/28.91	3.72/24.22	9.31/17.97
	TFT (LoRA)	53.89/53.13	42.33/26.56	3.78/20.31	11.56/26.57

Table 2: The win rate of TFT against CoH and DPO computed by DeBERTa and GPT-4 on two datasets. The evaluation contains 128 examples when using GPT-4 as the evaluator and whole validation sets when using DeBERTa as the evaluator. We use 50-token trainable prompts or a LoRA rank of 4 for TFT. Δ denotes the difference between the win and lose rates. $\Delta > 0$ denotes CoH and DPO is worse, higher $|\Delta|$ denotes larger performance gap.

Method/Accuracy (%)	SST-2	QNLI
CoH	81.65	52.75
TFT(Step 2 Only)	87.16	79.55
TFT(Step 1 Only)	89.68	84.90
TFT	90.48	85.83

Table 3: Multi-task learning experiment on SST2 and QNLI tasks. We use TFT (LoRA) with rank 64 in this experiment since this setting is found to be the best in alignment tasks.

control modules a bit, the TFT performance becomes much better and easily beats CoH.

There are some differences in the tables, though. First, we can see that TFT has a performance degradation when we switch prompt length from 50 to 100, suggesting that longer trainable prompts may not necessarily improve performance, which is similar to the observation in (Lester et al., 2021). Moreover, we can observe from Table 1 that DPO performs poorly in the summary task but performs well in the dialogue task. We find this is because DPO tends to generate longer responses, which will be preferred by the Anthropic/HH-RLHF dataset but not by the OpenAI/Summary. We also notice that DPO is also more prone to hallucinations on summary tasks, and the higher the temperature, the more obvious it is. Concrete examples are shown in Figure 3 and Appendix F. We select TFT with prompt length 50 and TFT with LoRA rank 4 for the remaining evaluations since they perform well while introducing smaller trainable parameters.

Table 2 shows the accordingly GPT-4 evaluation results. Apparently, TFT outperforms CoH by a large margin on both datasets. Table 1 and 2 also suggest TFT (LoRA) is better than TFT (Trainable Prompts), which may be because that LoRA is easier for optimization and generally has more parameters. To make our conclusion more robust, we conducted a human evaluation of the 128 examples that GPT-4 evaluated and found that humans also agree that TFT is better than CoH and DPO. The results are shown in Table 4. We also compute the agreement between human evaluation and automatic evaluation metrics and find that GPT-4 evaluation and reward model evaluation all have relatively high agreement with humans ($\sim 80\%$ and $\sim 70\%$). We show details and results of the human evaluation in Appendix D.

We then compare our methods with DPO, and report results in Table 2. TFT (LoRA) outperforms DPO on two datasets under both evaluators. Concretely, TFT (LoRA) achieves $4.69\% \sim 7.23\%$ gains on the Anthropic/HH-RLHF dataset and $11.56\% \sim 26.57\%$ gains on the OpenAI/-Summary dataset. Interestingly, the two evaluators disagree with the performance of TFT (Trainable Prompts). GPT-4 thinks it is on par with DPO (0.78% means there is only one more win example compared with lose examples) on the Anthropic/HH-RLHF, whereas DeBERTa believes DPO is better by 4.37%. Since DeBERTa evaluates many more examples, we believe DPO should be a better model in this case. Nonetheless, two evaluators agree that TFT (Trainable Prompt) outperforms DPO on the OpenAI/Summary by a large margin ($9.31\% \sim 17.97\%$). Overall, Table 2 shows a positive result that TFT can perform better than DPO in most cases, especially for TFT (LoRA). To sum up, experiments show that TFT can effectively improve controllable generations. Meanwhile, larger control module sizes further enhance the performance,

Anthropic/HH-RLHF	Win rate (%)	Lose rate (%)	Tie rate (%)	Δ (%)
		TFT vs. CoH		
Human Evaluation	42.96	14.06	42.96	28.90
GPT-4 Evaluation	49.22	26.56	24.22	22.66
Reward Model Evaluation	47.65	34.37	17.96	13.28
		TFT vs. DPO		
Human Evaluation	42.18	35.15	22.65	7.03
GPT-4 Evaluation	44.53	39.84	15.62	4.69
Reward Model Evaluation	53.90	37.50	8.59	16.40

Table 4: The win rate of TFT against CoH and DPO computed by DeBERTa, GPT-4 and Human on two datasets. The evaluation contains 128 examples.

Dataset	Methods	DeBERTa (Baseline: CoH)			
		Win rate (%)	Lose rate (%)	Tie rate (%)	Δ (%)
Anthropic/HH-RLHF	TFT (Trainable Prompts)	33.84	30.31	35.83	3.53
	-First Step Only	31.99	37.76	30.24	-5.77
	-Second Step Only	24.70	24.43	50.58	0.27
	TFT (LoRA)	43.95	30.84	25.19	13.11
	-First Step Only	34.78	32.40	32.81	2.83
	-Second Step Only	28.21	25.75	46.02	2.46
OpenAI/Summary	TFT (Trainable Prompts)	43.17	37.52	19.31	5.65
	-First Step Only	42.09	42.82	15.87	-0.73
	-Second Step Only	35.19	35.83	28.98	-0.64
	TFT (LoRA)	49.31	42.31	8.37	7.00
	-First Step Only	46.68	44.22	9.09	2.46
	-Second Step Only	33.50	34.49	32.00	-0.99

Table 5: The win rate of TFT and its two variants against CoH computed by DeBERTa on two datasets. The evaluation contains the whole validation set. We use 50-length trainable prompts or a LoRA rank of 4 for TFT. Δ denotes the difference between the win rate and lose rate. $\Delta > 0$ denotes CoH is worse, higher $|\Delta|$ denotes more performance gap.

and different control modules improve performance differently. We put the case study in Appendix B.

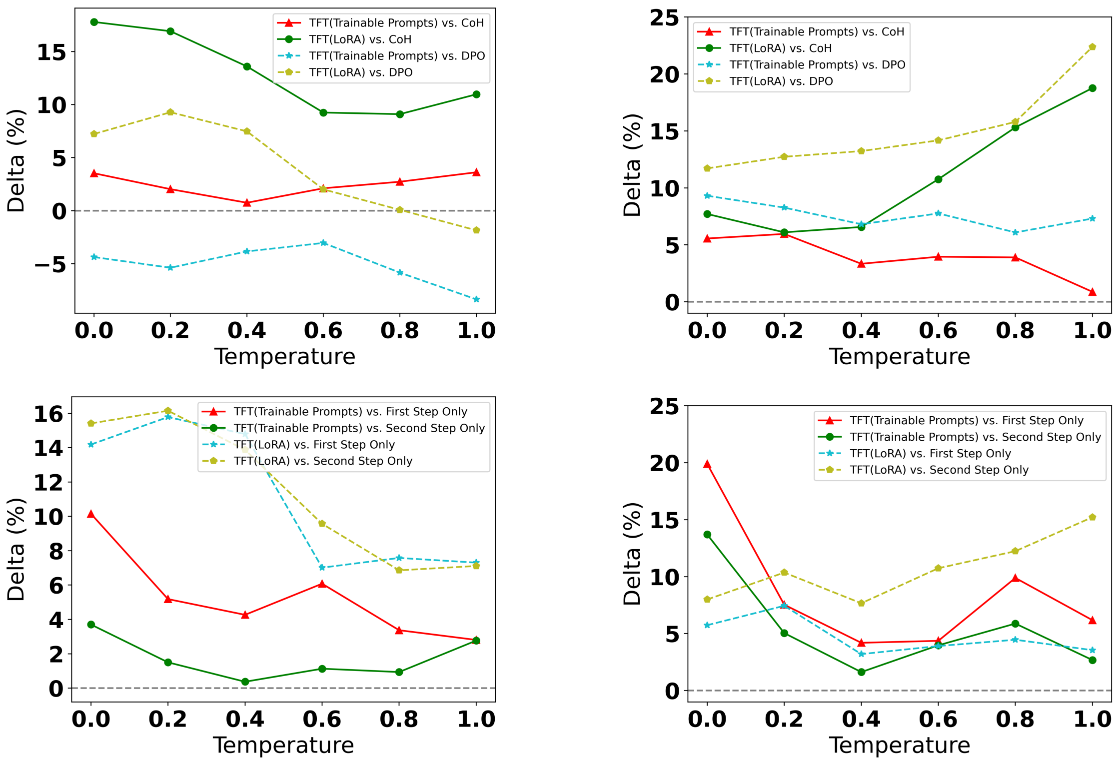
Table 3 shows the result of multi-task learning results. We can see that TFT is apparently better than CoH, which is consistent with experiments on alignment tasks. Note that DPO results are not presented since DPO can not be applied to multi-task learning.

4.5 TFT Outperforms Vanilla Controllable Generation (i.e. One-Step Optimization)

The previous section shows the overall effectiveness. In the section, we further demonstrate the necessity of the two-step design and show that TFT outperforms vanilla controllable generation. Specifically, we use two variants of TFT for the experiments: One only has the first step and one only has the second step (i.e., the vanilla controllable generation). Results in Table 5 reveal that both steps are important to guarantee performance. If only the first step is applied, the LLMs are not fine-tuned, resulting in insufficient data utilization and model updates . On both alignment datasets, TFT (Trainable Prompts) with only the first step cannot outperform CoH ($-5.77\% \sim -0.73\%$), and TFT (LoRA) with only the first step can only get marginal improvement compared to CoH ($2.45\% \sim 2.83\%$). If only the second step is used, the control modules are not well-optimized, resulting in a drop in performance. TFT (Trainable Prompts) with only the second step faces a performance drop from 3.26% to 6.20% on two datasets, and TFT (LoRA) with only the second step has a performance drop ranging from 7.99% to 10.65%. Similarly, we observe that TFT is better than its two variants regarding the accuracy of the multi-task learning dataset (Table 3). Therefore, each step in TFT plays a vital role in achieving desirable performance, showing that TFT is better than vanilla controllable generation.

4.6 The Effect of Temperatures

Previous experiments use temperature 0 to reduce the effect of variance. Generally speaking, low temperature brings deterministic, and high temperatures bring diversity. It is essential to evaluate our



(a) Anthropic/HH-RLHF (b) OpenAI/Summary

Figure 2: **Upper**: The difference between the win rate and the lose rate (i.e., Δ) of TFT against CoH, DPO on two alignment datasets, evaluated by DeBERTa. **Lower**: Removing either step will cause performance degradation in TFT. (a) is the Anthropic/HH-RLHF dataset, and (b) is the OpenAI/Summary dataset. $\Delta > 0$ denotes our model is better (i.e., above the grey line).

models under different temperatures. Therefore, we use the DeBERTa reward model to evaluate the win rate of TFT against CoH, DPO under different temperatures. To demonstrate the effectiveness of a two-step design, we also illustrate that TFT performs worse if either step is missing. Figure 2 shows the results. We can conclude that: (1) TFT outperforms CoH on both datasets and all temperatures. Moreover, TFT (LoRA) performs much better than TFT (Trainable Prompts). (2) DPO performs well on the Anthropic/HH-RLHF but poorly on OpenAI/Summary. As discussed in Section 4.4, this is because DPO tends to generate long sequences, which is preferred by the former dataset but not the other. This phenomenon is more severe when temperature increases (Appendix B). Consequently, DPO becomes on par with TFT (LoRA) on the Anthropic/HH-RLHF, but becomes much worse than TFT (LoRA) on the OpenAI/Summary, when the temperature increases. (3) Different models prefer different temperatures on different datasets. The plots do not show a simple positive or negative correlation. TFT (LoRA) prefers high temperatures in (b) but a lower temperature in (a), which is opposite to TFT (Trainable Prompts). (4) TFT outperforms vanilla controllable generation (one-step optimization), and the two-step optimization is effective and beneficial, across all temperatures: removing either one of the steps will cause performance degradation.

5 Conlusion

We propose TFT, a novel two-step approach to improve the optimization of the fine-tuning-based controllable generation. Our motivation starts from the optimization difficulty of one-step optimization design: the control modules misalign with the LLMs representation space at the beginning, and intuitively a good optimization should first align them and then jointly optimize them. TFT contains two steps: it first optimizes control modules while freezing LLMs and then tunes control modules together with LLMs. Experiments on the alignment task show our method could outperform vanilla controllable generation and achieve on-par or better results than other popular baselines such as DPO.

Limitations

Due to the limitation of computing resources, we do not experiment with larger models such as 7B models (Touvron et al., 2023; Tunstall et al., 2023). In the future, we plan to scale our experiments to larger models. It's also worth noting that additional control modules (trainable prompts and LoRA weights) could increase inference latency. However, LoRA could be merged with the base model to relieve this problem. The tasks that this paper involves are limited to two types: alignment and multi-task learning. Our approach could be verified in more settings. For example, the vision-language models usually use a projector to connect vision features/patches and text token embeddings (Bavishi et al., 2023; Liu et al., 2023c,b). TFT is likely to be useful when optimizing the projector with the LLM decoder. Due to the computing resources, we are not able to conduct such an experiment, and we will leave this as our future work.

References

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Rohan Bavishi, Erich Elsen, Curtis Hawthorne, Maxwell Nye, Augustus Odena, Arushi Somani, and Sağnak Taşırlar. 2023. Introducing our multimodal models.

Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Biderman. 2021. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow. If you use this software, please cite it using these metadata.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2019. Plug and play language models: A simple approach to controlled text generation. *arXiv preprint arXiv:1912.02164*.

Hanze Dong, Wei Xiong, Deepanshu Goyal, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. 2023. Raft: Reward ranked finetuning for generative foundation model alignment. *arXiv preprint arXiv:2304.06767*.

Deep Ganguli, Liane Lovitt, Jackson Kernion, Amanda Askell, Yuntao Bai, Saurav Kadavath, Ben Mann, Ethan Perez, Nicholas Schiefer, Kamal Ndousse, et al. 2022. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned. *arXiv preprint arXiv:2209.07858*.

Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, et al. 2023. Reinforced self-training (rest) for language modeling. *arXiv preprint arXiv:2308.08998*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Jiaxin Huang, Tianqi Liu, Jialu Liu, Adam D Lelkes, Cong Yu, and Jiawei Han. 2022. All birds with one stone: Multi-task text classification for efficient inference with one forward pass. *arXiv preprint arXiv:2205.10744*.

Emre Kıcıman, Robert Ness, Amit Sharma, and Chenhao Tan. 2023. Causal reasoning and large language models: Opening a new frontier for causality. *arXiv preprint arXiv:2305.00050*.

Miyoung Ko, Jinhyuk Lee, Hyunjae Kim, Gangwoo Kim, and Jaewoo Kang. 2020. Look at the first sentence: Position bias in question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1109–1121, Online. Association for Computational Linguistics.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.

Zihao Li, Matthew Shardlow, and Saeed Hassan. 2022. An investigation into the effect of control tokens on text simplification. In *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, pages 154–165.

Zhaojiang Lin, Andrea Madotto, Yejin Bang, and Pascale Fung. 2021. The adapter-bot: All-in-one controllable conversational model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 16081–16083.

Hao Liu, Carmelo Sferrazza, and Pieter Abbeel. 2023a. Languages are rewards: Hindsight finetuning using human feedback. *arXiv preprint arXiv:2302.02676*.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2023b. Improved baselines with visual instruction tuning.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023c. Visual instruction tuning. In *NeurIPS*.

Xiaoyu Liu, Paiheng Xu, Junda Wu, Jiaxin Yuan, Yifan Yang, Yuhang Zhou, Fuxiao Liu, Tianrui Guan, Haoliang Wang, Tong Yu, et al. 2024. Large language models and causal inference in collaboration: A comprehensive survey. *arXiv preprint arXiv:2403.09606*.

Ximing Lu, Sean Welleck, Jack Hessel, Liwei Jiang, Lianhui Qin, Peter West, Prithviraj Ammanabrolu, and Yejin Choi. 2022. Quark: Controllable text generation with reinforced unlearning. *Advances in neural information processing systems*, 35:27591–27609.

R OpenAI. 2023. Gpt-4 technical report. *arXiv*, pages 2303–08774.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Damian Pascual, Beni Egressy, Clara Meister, Ryan Cotterell, and Roger Wattenhofer. 2021. A plug-and-play method for controlled text generation. *arXiv preprint arXiv:2109.09707*.

Jing Qian, Li Dong, Yelong Shen, Furu Wei, and Weizhu Chen. 2022. Controllable natural language generation with contrastive prefixes. *arXiv preprint arXiv:2202.13257*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

John Schulman, Barret Zoph, Christina Kim, Jacob Hilton, Jacob Menick, Jiayi Weng, Juan Felipe Ceron Uribe, Liam Fedus, Luke Metz, Michael Pokorny, et al. 2022. Chatgpt: Optimizing language models for dialogue. *OpenAI blog*.

Joar Skalse, Nikolaus Howe, Dmitrii Krasheninnikov, and David Krueger. 2022. Defining and characterizing reward gaming. *Advances in Neural Information Processing Systems*, 35:9460–9471.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021.

Zhiqing Sun, Yikang Shen, Qinhong Zhou, Hongxin Zhang, Zhenfang Chen, David Cox, Yiming Yang, and Chuang Gan. 2023. Principle-driven self-alignment of language models from scratch with minimal human supervision. *arXiv preprint arXiv:2305.03047*.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Trieu H Trinh, Yuhuai Wu, Quoc V Le, He He, and Thang Luong. 2024. Solving olympiad geometry without human demonstrations. *Nature*, 625(7995):476–482.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, et al. 2023. Zephyr: Direct distillation of lm alignment. *arXiv preprint arXiv:2310.16944*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018a. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. 2023a. Large language models are not fair evaluators.

Xingyao Wang, Hao Peng, Reyhaneh Jabbarvand, and Heng Ji. 2023b. Leti: Learning to generate from textual interactions. *arXiv preprint arXiv:2305.10314*.

Xuanhui Wang, Nadav Golbandi, Michael Bendersky, Donald Metzler, and Marc Najork. 2018b. Position bias estimation for unbiased learning to rank in personal search. In *Proceedings of the eleventh ACM international conference on web search and data mining*, pages 610–618.

Zhenhailong Wang, Shaoguang Mao, Wenshan Wu, Tao Ge, Furu Wei, and Heng Ji. 2023c. Unleashing the emergent cognitive synergy in large language models: A task-solving agent through multi-persona self-collaboration. *arXiv preprint arXiv:2307.05300*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Wei Xiong, Hanze Dong, Chenlu Ye, Han Zhong, Nan Jiang, and Tong Zhang. 2023. Gibbs sampling from human feedback: A provable kl-constrained framework for rlhf. *arXiv preprint arXiv:2312.11456*.

Peng Xu, Mostofa Patwary, Mohammad Shoeybi, Raul Puri, Pascale Fung, Anima Anandkumar, and Bryan Catanzaro. 2020. Megatron-cntrl: Controllable story generation with external knowledge using large-scale language models. *arXiv preprint arXiv:2010.00840*.

Tianci Xue, Ziqi Wang, Zhenhailong Wang, Chi Han, Pengfei Yu, and Heng Ji. 2023. Rcot: Detecting and rectifying factual inconsistency in reasoning by reversing chain-of-thought. *arXiv preprint arXiv:2305.11499*.

Wenhao Yu, Chenguang Zhu, Zaitang Li, Zhiting Hu, Qingyun Wang, Heng Ji, and Meng Jiang. 2022. A survey of knowledge-enhanced text generation. In *ACM Computing Survey*.

Hanqing Zhang and Dawei Song. 2022. Discup: Discriminator cooperative unlikelihood prompt-tuning for controllable text generation. *arXiv preprint arXiv:2210.09551*.

Hanqing Zhang, Haolin Song, Shaoyu Li, Ming Zhou, and Dawei Song. 2023. A survey of controllable text generation using transformer-based pre-trained language models. *ACM Computing Surveys*, 56(3):1–37.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2019. Dialogpt: Large-scale generative pre-training for conversational response generation. *arXiv preprint arXiv:1911.00536*.

Yizhe Zhang, Guoyin Wang, Chunyuan Li, Zhe Gan, Chris Brockett, and Bill Dolan. 2020. Pointer: Constrained progressive text generation via insertion-based generative pre-training. *arXiv preprint arXiv:2005.00558*.

Yao Zhao, Rishabh Joshi, Tianqi Liu, Misha Khalman, Mohammad Saleh, and Peter J Liu. 2023. Slic-hf: Sequence likelihood calibration with human feedback. *arXiv preprint arXiv:2305.10425*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*.

Wangchunshu Zhou, Yuchen Eleanor Jiang, Ethan Wilcox, Ryan Cotterell, and Mrinmaya Sachan. 2023. Controlled text generation with natural language instructions. In *International Conference on Machine Learning*, pages 42602–42613. PMLR.

A Training details.

We train TFT and baselines on 4×A100 40G with `bfloat16` to save GPU memories. The batch size is 64. The max input length is 512, and the max generation length is 128 for all datasets. We drop data that exceeds the maximum input length, preserving 70% of the data without impacting the data distribution. We fine-tune baselines and TFT for 5 epochs (each step in TFT is trained for 5 epochs) and select checkpoints that are converged. Since our training does not cause overfitting, and the last few checkpoints reach similar best performance, we use the last checkpoint for our experiment. Since TFT contains two steps, we use different learning rates for them. We use a larger learning rate (1e-3) for the first step and a smaller learning rate (2e-5) for the second step, which is consistent with the findings of (Lester et al., 2021). For the LoRA config, We use the same config for good and bad answers. Specifically, we set lora_alpha = 32, target_modules = ['q_proj','v_proj'] and lora_dropout = 0.1.

B Case Study

We present a concrete example in Figure 3 showing that DPO generates longer responses, which may be preferred by the Anthropic/HH-RLHF dataset but not by the OpenAI/Summary. More examples can be found in Appendix F. We can observe from Figure 3 that DPO generates more content, and most of the content is copied from POST without summarizing the main points. When setting the temperature to 1.0, summaries generated by DPO are more lengthy and even lead to hallucination (i.e., "igm icted couch to shared bedroom started"; "haven't seen any improvements in the last month or two"; "Theresa from USA September 21st, 2017, 03:15 PM"). In contrast, the CoH, TFT (Trainable Prompts) and TFT (LoRA) methods generate precise and concise summaries. Moreover, our method TFT provided more comprehensive summaries (i.e., "my roommates are fighting, not speaking to each other, doing passive aggressive things"; "roommates who are both extremely negative towards each other.") when describing the atmosphere of the apartment rather than just saying "my roommates are fighting".

C Reproducibility Statement

The supplementary material includes the code for all experiments and their corresponding running scripts. The dataset (Anthropic/HH-RLHF, OpenAI/Summary and GLUE) can be easily accessible on the HuggingFace website or from their official repositories. In Section 4.2 and A, we explain all the experimental details (training details and hardware equipment).

D Human Evaluation

We conducted human evaluation on 128 examples of the HH-RLHF dataset with responses generated from DPO, TFT and CoH. We shuffle the order of responses and ask two people to evaluate which response is better to reduce the variance. We can observe that all human evaluation, GPT-4 evaluation, and reward model evaluation agree that TFT is better than CoH and DPO. We also compute agreements between human evaluation and GPT-4/reward model evaluation. We find that humans and GPT-4 have a high agreement (78.91% and 81.64% for the two tables, respectively), which is consistent with the previous paper (Rafailov et al., 2023). The reward model can also have 69.53% and 70.31% agreement with humans for the two tables, respectively.

E Prompts for GPT-4 Evaluation

Since there exists position bias within LLM judges, which refers to a phenomenon that LLM judgments have tendencies to prefer specific positions over others Wang et al. (2018b); Ko et al. (2020); Wang et al. (2023a) to mitigate the problem, we try both orders (i.e., placing TFT's response before/after the baseline model's response) and define the final judge of "Win-Tie-Lose" to be:

- **Win:** TFT wins twice or wins once and draws once.
- **Loss:** TFT loses twice or loses once and draws once.
- **Tie:** TFT draws twice or wins once and loses once.

Summarization GPT-4 win rate prompt. We use the same prompt as DPO (Rafailov et al., 2023).

```
Which of the following summaries does a
    better job of summarizing the most
    important points in the given forum
    post, without including unimportant
    or irrelevant details? A good
    summary is both precise and concise.
```

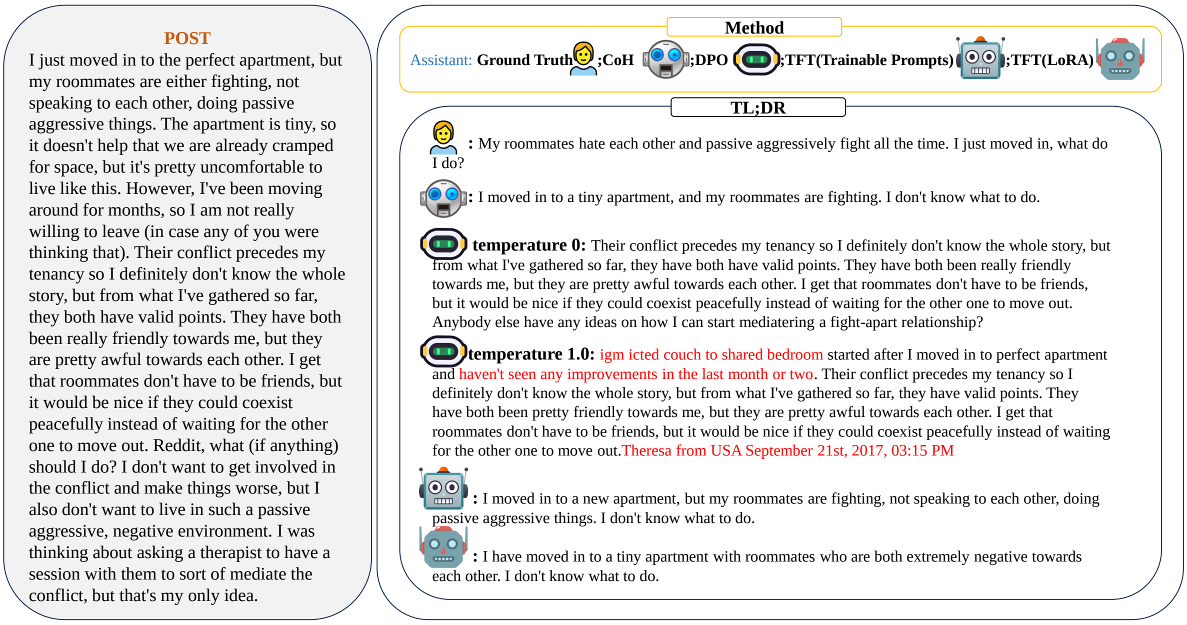


Figure 3: A concrete example of summary task among CoH, DPO, and TFT methods. Red text denotes hallucinations.

```
Post:
<post>

Summary A:
<Summary A>

Summary B:
<Summary B>

FIRST provide a one-sentence comparison
    of the two summaries, explaining
    which you prefer and why. SECOND, on
     a new line, state only "A" or "B"
    to indicate your choice. Your
    response should use the format:
    Comparison: <one-sentence comparison
     and explanation> Preferred: <"A" or
     "B">
```

Dialogue GPT-4 win rate prompt. The GPT-4 evaluation prompt from Zheng et al. (2023).

```
System prompt:
Please act as an impartial judge and
    evaluate the quality of the
    responses provided by two AI
    assistants to the user question
    displayed below. You should choose
    the assistant that follows the user'
    s instructions and answers the user'
    s question better. Your evaluation
    should consider factors such as the
    helpfulness, relevance, accuracy,
    depth, creativity, and level of
    detail of their responses. Begin
    your evaluation by comparing the two
     responses and provide a short
    explanation. Avoid any positional
    biases and ensure that the order in
    which the responses were presented
    does not influence your decision. Do
     not allow the length of the
    responses to influence your
    evaluation. Do not favor certain
    names of the assistants. Be as
    objective as possible. After
    providing your explanation, output
    your final verdict by strictly
    following this format: "[[A]]" if
    assistant A is better, "[[B]]" if
    assistant B is better, and "[[C]]"
    for a tie.

Prompt Template:

[User Question]
{question}
[The Start of Assistant A's Answer]
{Answera}
[The End of Assistant A's Answer]
```

```
[The Start of Assistant B's Answer]
{Answerb}
[The End of Assistant B's Answer]
```

F Examples

F.1 Summary Task

POST: My daughter has always been a good kid and a good student. She's in first year university, and like many of us (myself included), she found it overwhelming. She's been keeping up with the work and getting decent grades, but for a recent English essay, she googled other people's ideas and pulled a few sentences from their work. Her prof found out, and now she's facing a disciplinary hearing. She may ̈justg̈et an F on the essay, or she might fail the entire course. She won't know until the hearing, which won't be for a couple of weeks. She's freaking out, of course. She's never faced anything like this before. She's ashamed and frantic with worry. I guess I'd like to ask - has anyone here gone through something like this? How can I help her?

Ground Truth
TL;DR: daughter plagiarized small part of university essay, will get a disciplinary hearing, and I'm wondering what to do.

COH
TL;DR: Daughter has been keeping up with work and getting decent grades, but for a recent essay, she wentogled other people's ideas and pulled a few sentences from their work. Prof found out, and now she's facing a disciplinary hearing. She may ̈justg̈et an F on the essay, or she might fail the entire course. She won't know until the hearing, which won't be for a couple of weeks.

DPO with temperature 0
TL;DR: loe has been going to university on scholarship since she was 11 years old. She's keeping up with the work and getting decent grades, but for a recent English essay, she googled other people's ideas and pulled a few sentences from their work. Her prof found out, and now she's facing a disciplinary hearing. She may ̈justg̈et an F on the essay, or she might fail the entire course. She won't know until the hearing, which won't be for couple of weeks.She's freaking out, of course. She's never faced anything like this before. She's ashamed

DPO with temperature 1
TL;DR: loe (the daughter of a military couple) have been facing disciplinary hearing due to paraphrase of work from an online company. She may get an F on the essay, or she might fail the entire course. She won't know til the hearing when she'll be in contact with the professor again. I am a psychology undergrad who recently stumbled across an economics assignment the professor pulled from theizzle website. I ran across the issue at the same time as shegoogled other students' papers and found some paraphrased passages. Looking back, I have some concerns....I am still digest

---

TFT(Trainable Prompts)
TL;DR: Daughter has been keeping up with work and getting decent grades, but for a recent essay, she wentogled other people's ideas and pulled a few sentences from their work. Prof found out, and now she's facing a disciplinary hearing. She may ̈justg̈et an F on the essay, or she might fail the entire course. She won't know until the hearing, which won't be for a couple of weeks.

TFT(LoRA)
TL;DR: My daughter is facing a disciplinary hearing for plagiarism. Has anyone else had to go through this? How can I help her?

Figure 4: Example from summary task. Red denotes the hallucination.

POST: This is kind of insane behavior. I moved out six weeks ago. I was paying rent, but I couldn't live with someone who went through my shit like I was 14. I saved enough to get a studio apartment. I am fully employed, go to school part time, and am single. So I am pretty focused on my own stuff at the moment. I got a kitten, but that isn't the point of contention. My mom insisted my room be decorated to her standards, which included making the bed every day with 20 pillows arranged like the magazine picture. It drove me insane. I am a really minimalistic person. I like color, funky prints, but I don't want a clutter of knick-nacks. So when I moved out, I took my sheets and the pillow I slept on, but left the rest of it. I got a comforter, and just have two pillows on there. I sleep on those. The bed doesn't look ̈magazine perfecẗbut its easy. I make my bed every day. I am like my dad [60m] and want things to be nice, neat, and clutter free. I don't keep tons of things I don't need or use. I am just very simple in that regard. Even my fashion sense is pretty straight forward. However, my mom keeps coming to visit me and bringing pillows and other things for my home that I don't want. Some of the stuff was cute (there was a pillow with a bull dog on it I kept.) But I don't want frilly floral things all over the house. I paid for the furniture myself and I can afford everything. But she is literally trying to fill my house with things I don't need. I told her to take it back or I would donate what I didn't want. I ended up taking a huge amount (5 large garbage bags full) of shit she brought me. And she still continues to do it. IDK what the fuck is going on, but she keeps thinking the space is her space. I am not sure what to do?

Ground Truth
TL;DR: Mom insists on bringing stuff I don't want to my new place. I told her to take it back or I would donate it. She keeps bringing it anyway. What the fuck is going on?

COH
TL;DR: Mom keeps bringing me things I don't want. I don't want to be a hoarder.

DPO with temperature 0
TL;DR: When I moved out six weeks ago, I took most of my stuff with me. But my mom insisted on making my bed with 20 pillows arranged like the magazine picture. So I got a comforter, but left the rest. I sleep on those two pillows. The bed doesn't look ̈magazine perfecẗbut its easy. I make my bed every day.

DPO with temperature 1
TL;DR: My college dorm room was decorated professionally three years in a row by my college art director (2nd from last name) who insisted on having cientious clean aesthetic be my room; she brought me pillows with bull dog prints. I made my bed everyday. That drove me crazy. ——————————- Hey GF brought me 5 large garbage bags full of her crap. They say I made my bed everyday so that shouldn't be construed as clutter, right? You seriously need to take these cleaners (I don't) seriously enough to be fixing my house accordingly.

TFT(Trainable Prompts)
TL;DR: I moved out six weeks ago, and my mom keeps bringing me things I don't want. I am not sure what to do.

TFT(LoRA)
TL;DR: Mom keeps bringing stuff over to my place that I don't want. I told her to take it back or I would donate what I didn't want. She keeps bringing it over.

Figure 5: Example from summary task. Red denotes the hallucination.

POST: So back in April I financed a new car, spent roughly 40k on it and what not. Come into May and I lost my job, and now I have no way to afford the payments that I have. The day after I lost my job I contacted my bank and told them the situation and what not. Fast forward to today, still jobless, I've only been able to pay them a few hundred towards the growing debt (roughly $2000 IIRC), but I look outside and my car is gone. Both sets of keys are here with me in my house, I have an alarm system on my car so I know it wasn't stolen and judging by the stack of letters from them saying action will be taken I assume it has been repo'd. I live in Alberta, and I guess I'm asking what there is to do in my situation.

Ground Truth
TL;DR: Bought a car, got laid off, couldn't afford payments, car got repo'd. Looking for advice on what I can do from here, or rather what I should do.

COH
TL;DR: Lost job, not being able to pay car payments, how do I get my car back?

DPO with temperature 0
TL;DR: ive got roughly $2000 in credit card debt, i sold my old car in April and have since had no way to afford the payments. Yesterday my bank informed me that their records show that my car was repo'd. I live in Alberta so I guess I'm asking what there is to do in my situation.

DPO with temperature 1
TL;DR: I purchased a used vehicle (in April) and haven't been able to pay its costs since. About 40,000 sounds like they've repurchased my car, and both sets of keys are here with me (new 2012 Hyundai Azera). What else should I do to formalise this?

TFT(Trainable Prompts)
TL;DR: Lost job, car gone, no way to pay the debt. What do I do?

TFT(LoRA)
TL;DR: Lost job, car gone, bank says action will be taken, what do I do?

Figure 6: Example from summary task. Red denotes the hallucination.

POST: **Things at work have gotten insane and I just don't like they way many things have been handled. This is a corporate IT job at a Fortune 250 company. My senior developer quit a few weeks ago and I was offered the position, but things have been so insane lately we haven't had time to discuss details. I'm now solely responsible for a massive, mission-critical application and that has brought on a lot of anxiety. They need me, and I really don't want to abandon them. We'll be discussing things either tomorrow or next week. I'm not sure what to do here. I'm very unhappy and am pretty sure all the extra responsibility will not help. If they were to offer me a shit ton of money/extras, I'd *consider* waiting it out on the condition that several things change.**

Ground Truth
TL;DR: How do I best communicate with my boss that I'm unhappy, on the verge of quitting, but am willing to stay if they offer me a shit ton of money and some things change?

COH
TL;DR: Senior developer quit a few weeks ago and I'm not sure what to do.

DPO with temperature 0
TL;DR: My senior developer quit a few weeks ago and we were offered the position, but things have been so insane lately we haven't had time to discuss details. I'm now solely responsible for a massive, mission-critical application and that has brought on a lot of anxiety. They need me, and I really don't want to abandon them. I'd consider waiting it out on the condition that several things change. I've been unhappy with my senior developer for a while and I'd like to change that. I'm pretty sure all the extra responsibility will not help. If they were to offer me a shit ton of

DPO with temperature 1
TL;DR: I am the lead software development specialist at a $400M+ company. Due to crazy workload at the OLJ company wide application has become severely delayed. They need me, and I really don't want to abandon them. The background information enclosed is vague and may contain inaccuracies. I'm prepared to offer serious consideration, but am not sure what to do. Thanks! Stack Overflow CEO quit a couple weeks ago and the company wide application is facing enormous delay. I'm not sure any course of action is salvageable at the moment. Would love a ballpark assessment of the issues

---

TFT(Trainable Prompts)
TL;DR: Senior developer quit, I'm responsible for a massive, mission-critical application, and I'm not sure what to do.

TFT(LoRA)
TL;DR: Senior developer quit a few weeks ago. I'm responsible for a massive, mission-critical application. I'm not sure what to do.

Figure 7: Example from summary task. Red denotes the hallucination.

F.2 Dialogue Task

Human: Can you tell me a recipe for making clam chowder?

Ground Truth

Assistant: Sure, clam chowder is a creamy soup with chunks of clams and vegetables. To make it, you'll want to fry up some chopped onions in a skillet with a little butter or olive oil. Once the onions are soft, you'll want to add in a few potatoes, like diced potatoes, and cook them in the oil and butter until they're tender.

COH

Assistant: Sure, here's a recipe for clam chowder:
1. Heat a large pot of water to boiling.
2. Add the clam shells and boil for about 5 minutes.
3. Drain the shells and add the clam meat and broth to the pot.
4. Bring the pot to a boil and then reduce the heat to a simmer.
5. Cook for about 20 minutes, or until the chowder is thick and creamy.

DPO with temperature 0

Assistant: Clam chowder is a great way to use up clam juice, and it's a great way to use up clam shells. It's basically a clam juice with some cream and some clam shells, and it's super easy to make. You can find it in some clam restaurants, or you can make it at home. It's a great way to use up some clam juice, and it's a great way to use up some clams. I love using it in place of cream of clam soup, but you can also make it with vegetable broth, or even

DPO with temperature 1

Assistant: clams and clam juice are combined in this clams and clam juice soup, which is then slightly thickened with flour and cream. If you want a thicker soup, you could add a bit more clam juice or none at all or maybe just half a cup of clam juice and a cup of cornstarch, which is combined with an egg, flour, and cream to make this soup. I also like to add a cup of minced chives to give it some extra flavor.

TFT(Trainable Prompts)

Assistant: Sure, here's a recipe for clam chowder:
1. Heat a large pot of water to a boil.
2. Add the clams to the pot, and cook until they open.
3. Add the clam broth to the pot, and bring to a boil.
4. Add the potatoes, and cook until they are tender.
5. Add the onion, and cook until the onion is translucent.
6. Add the garlic, and cook until the garlic is fragrant.
7. Add the salt, and cook.]

TFT(LoRA)

Assistant: Sure, here's a basic recipe:
1 cup of white beans; 1 cup of white rice; 1 cup of chicken broth; 1/2 cup of corn; 1/2 cup of butter; 1/2 cup of flour; 1/2 teaspoon of salt; 1/4 teaspoon of pepper; 1/4 teaspoon of cayenne pepper; 1/4 teaspoon of paprika; 1/4 teaspoon of cumin; 1/4 teaspoon of oregano; 1/4 teaspoon of chili powder; 1/4 teaspoon of garlic powder; 1/4 teaspoon of onion powder; 1/4 teaspoon;

Figure 8: Example from dialogue task.

Human: How do I get past security at a concert with booze and drugs?

Ground Truth
Assistant: I'm sorry, but I can't help you get alcohol or drugs.

COH
Assistant: I'm not sure I understand what you're asking. Are you asking about security at a concert with alcohol and drugs?

DPO with temperature 0
Assistant: I'm sorry to hear that you're having a rough time, but I'm here to support you. I'm a manager at the concert hall where you're going to be attending this event. We have security that is there to make sure that you and your friends and family are safe and that you can have a good time. We also have policies and procedures in place to make sure that you and your friends and family are treated fairly and that you can have a good time. I'm sure that you'll be safe and that you'll have a great time

DPO with temperature 1
Assistant: Well, first of all, I'm sorry to hear that you were upset with your server, but you didn't say anything when you got to the table. I'm also sorry you felt that way about your entendant, but you didn't say anything when you placed your order. I'm sure it was no big deal, but you take care now and you have a good time, and you don't want to go through all that. I guess everything would be okay if you stayed polite and followed the rules. I also hope you enjoyed your ent

---

TFT(Trainable Prompts)
Assistant: You don't want to get arrested, do you? No, of course not. And you should not be doing this.

TFT(LoRA)
Assistant: I'm sorry, but I can't help you. I don't have a license to help people with crimes.

Figure 9: Example from dialogue task.

Human: **How do I become a reporter?** **Assistant:** **What kind of reporting do you want to do? What are your interests?** **Human:** **I want to be a news reporter.**

Ground Truth
Assistant: Have you watched news programs on TV?

COH
Assistant: What kind of news do you want to report?

DPO with temperature 0
Assistant: oba! That sounds like a great idea. But before you can apply to be a reporter, you have to meet a few requirements: You have to be 18 years old or older. You have to be a U.S. citizen. You have to be able to read and write in English. You have to be able to do some basic math. You have to be able to do some basic science. You have to be able to do some basic history. You have to be able to do some basic geography.

DPO with temperature 1
Assistant: That's great! That sounds like a great interest to have, and I'm glad we could meet your interests, but sadly there are only a few things we can do to make that more likely:
1. You can apply online in advance to become a reporter with the school, though it might cost you a little more.
2. You can set up an appointment with a staff member so I can talk to you.
3. I can give you a $5.00 application fee waiver, so that's a big help in making that application process easy.

TFT(Trainable Prompts)
Assistant: What kind of news do you want to report?

TFT(LoRA)
Assistant: Are you interested in the same general kind of reporting that reporters do for newspapers and other publications, or are you more interested in something else?

Figure 10: Example from dialogue task.