MORL-Prompt: An Empirical Analysis of Multi-Objective Reinforcement Learning for Discrete Prompt Optimization

Anonymous ACL submission

Abstract

RL-based techniques can be used to search for prompts that when fed into a target language model maximize a set of user-specified reward functions. However, in many target applications, the natural reward functions are in tension with one another – for example, content preservation vs. style matching in style transfer tasks. Current techniques focus on maximizing the average of reward functions, which does not necessarily lead to prompts that achieve *balance across rewards* – an issue that has been well-studied in the multi-objective and robust optimization literature. In this paper, we adapt several techniques for multi-objective optimization to RL-based discrete prompt optimization – two that consider volume of the Pareto reward surface, and another that chooses an update direction that benefits all rewards simultaneously. We conduct an empirical analysis of these methods on two NLP tasks: style transfer and machine translation, each using three competing reward functions. Our experiments demonstrate that multi-objective methods that directly optimize volume perform better and achieve a better balance of all rewards than those that attempt to find monotonic update directions.

1 Introduction

Discrete prompt tuning involves refining a text prompt for a language model (LM) to maximize a set of user-specified objectives on the LM's output (Shin et al., 2020; Schick and Schütze, 2020; Wen et al., 2023). Successful techniques for prompt tuning allow users to control and adapt powerful LLMs to new tasks without the trial-and-error of manual prompt design. While RL-based techniques have been shown to be effective at finding prompts that optimize an average of rewards (Deng et al., 2022), in many target applications, the natural reward functions are in tension with one another.

For example, as depicted in Figure 1, many style transfer tasks need to both maintain content preservation while simultaneously maximizing transfer into the target style – two objectives that are directly at odds with one another. Thus, current techniques result in a phenomenon we refer to as *objective collapse*: focusing on maximizing the average of reward functions (also called *scalarization*) can lead to prompts that disproportionately maximize a subset of objectives at the expense of others. For instance, in Figure 1, the prompt on the left side tends to produce LM outputs (represented by blue dots) that prioritize one objective over the other. Conversely, the prompt on the right side produces samples that achieve reasonable performance across all objectives simultaneously. However, in both cases the *average reward* is nearly equivalent.

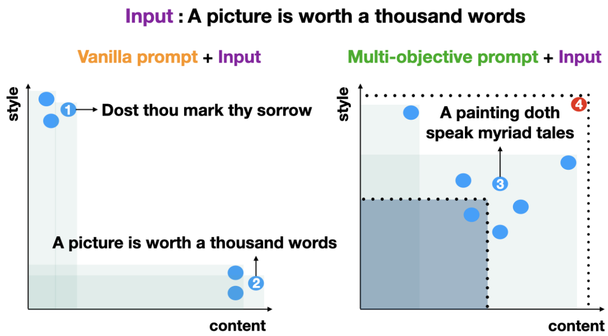


Figure 1: A modern to Shakespearean text style transfer setting where each dot represents an output sentence sampled from an LM conditioned on either a prompt trained with average reward (left) or a prompt trained using multi-objective optimziation techniques (right). The output sample 1 only optimizes for style match, while output sample 2 only addresses content preservation. Sample 3, on the other hand, balances both objectives at the same time. The shaded regions indicate measures of volume of the Pareto reward surface.

In this paper, we adapt several techniques for multi-objective optimization to the RL-based discrete prompt optimization setting and evaluate their effectiveness in achieving a more useful balance of rewards in downstream tasks. Specifically, we propose two approaches that consider the volume of the Pareto reward surface, and another that chooses an update direction that benefits all rewards simultaneously.

Our first method computes the hypervolume indicator (HVI) (Knowles et al., 2004) for a set of samples drawn from a given prompt, and treats this measure as the final reward in RL. Intuitively, HVI measures the area under the Pareto frontier of the outputs sampled from the current prompt (shown by the outer rectangular region in Figure 1). Samples that achieve a better balance of reward lift the Pareto frontier and increase HVI. However, this method has a potential pitfall: if even a single outlier sample (e.g. depicted by the red dot labeled with a four in Figure 1) happens to achieve a high value across all rewards, the HVI can be extremely high. This *dominant outlier effect* may reduce the robustness of HVI optimization in an RL setting. Thus, we also propose and evaluate a simpler method that approximates the expected volume by simply computing the average *product of rewards* (depicted by the dark rectangular region in Figure 1). Our final approach takes a different strategy based on *steepest gradient descent* (Fliege and Svaiter, 2000). Here, we approximate the gradient of the expectation of each individual reward separately, and then search for an update direction in their span that has a non-negative dot product with each reward gradient – i.e. designed to make monotonic progress in every reward simultaneously.

To understand the effectiveness of these approaches in the discrete prompt optimization setting, we conduct experiments on two text generation tasks: text style transfer and machine translation using sets of competing reward functions. Our findings indicate that volume-based methods are most effective in this setting, achieving substantial gains in balance compared to the baseline methods. RL-based steepest descent also improves balance, but not nearly as robustly as volume-based methods.

2 Problem Statement

Given multiple objectives and their corresponding reward functions $\{r_1, r_2, \dots r_m\}$, we propose a discrete prompt optimization method for controlled text generation. We refer to a set of discrete prompts as $Z = \{z_1, z_2, \dots z_n\}$, the input text as x, and the text generated by the LM as y. The unsupervised task requires texts as inputs whereas supervised tasks also take the targets as additional inputs. We aim to generate a prompt that is added to the beginning of the input and cause the LM to generate output text compliant with the objectives.

3 Methodology

3.1 Optimization problem

We formulate discrete prompt optimization as an RL problem, where we train a multi-layer perceptron (MLP) head over a frozen language model as our policy network.

At each step, given a text input x, we sample k prompts $\{z_1, z_2, \dots z_k\}$ from the policy π_θ, where θ represents the policy parameters. Subsequently, we utilize another frozen language model p_{LM} to generate $\hat{k}$ output sentences for each pair of input x and prompt z_i. Then, we assess the quality of these outputs using the reward function r_i corresponding to each objective[1]. Finally, the optimization problem centers on maximizing these rewards as follows:

$$\max_\theta \sum_{i=1}^{k\cdot\hat{k}} \mathbb{E}_{z\sim\pi_\theta}\left[\mathbb{E}_{y\sim p_{LM}(y|x,z)}\left[\sum_{j=1}^{m} r_j\left(y,x\right)\right]\right] \quad (1)$$

3.2 RL-based Volume Indicator

In this section, we investigate two approaches that aim to improve the volume coverage of rewards.

3.2.1 Hyper-volume indicator

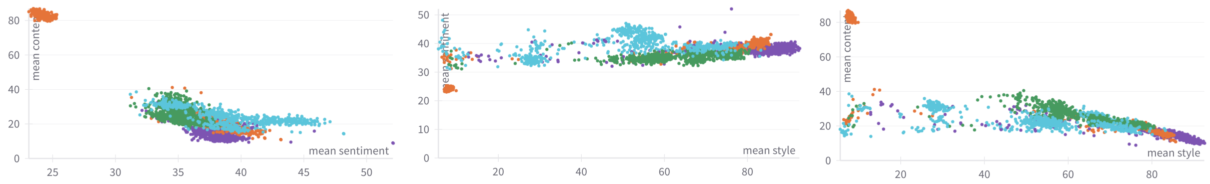
The hypervolume indicator (Knowles et al., 2004; Zitzler and Thiele, 1998) is defined for a point set $S \subset \mathbb{R}^d$ and a reference point $r \in \mathbb{R}^d$, where hypervolume indicator of S quantifies the region dominated by S and bounded by r, denoted as:

$$H(\mathrm{S})=\Lambda\left(\left\{q\in\mathbb{R}^d \mid \exists p\in \mathrm{S} : p\leq q \text{ and } q\leq r\right\}\right)$$

where $\Lambda(\cdot)$ shows the Lebesgue measure for the sub-space.

We consider the hypervolume indicator of the individual reward functions as the reward signal for training the policy network. We then calculate the efficient SQL learning loss (Guo et al., 2022) based on this reward and update the policy network's parameters using gradient descent.

[1]For simplicity, we assume the reward value is solely dependent on the generated text y and the input text x. It can be easily expanded to include prompt z or the reference text, if necessary.



Sentiment vs Content | Style vs Sentiment | Style vs Content

Figure 2: Text Style Transfer. From left to right, positive sentiment vs. content match, Shakespearean style vs. positive sentiment, and Shakespearean style vs. content match for different settings of average reward, hyper volume indicator reward, product reward, and multiple gradient descent algorithm are shown.

3.2.2 Expected product of rewards

We obtain $\hat{k}$ samples as output per prompt and for each sentence, we compute all m reward values, and calculate the product of rewards. We utilize the expected product of rewards across all $\hat{k}$ samples as the final reward signal for policy updates.

3.3 Multiple Gradient Descent Algorithm with RL

In this section, we investigate multiple gradient descent algorithm (MGDA), which performs the steepest descent for multi-criteria optimization (Fliege and Svaiter, 2000), where the goal is to find a direction d_t that improves all the objectives by the amount of α_t, at each step t.

$$
\begin{aligned}
&(d_t, \alpha_t) = \arg \min_{d \in \mathbb{R}^n, \alpha \in \mathbb{R}} \alpha + \frac{1}{2}\|d\|^2, \\
&\text{s.t. } \nabla \mathcal{L}_i\left(\theta_t\right)^T d \leq \alpha, \quad i = 1, \ldots, m.
\end{aligned} \tag{2}
$$

This approach has been used in continuous multi-task settings (Sener and Koltun, 2019; Lin et al., 2019). However, as we optimize for a discrete case, we compute stochastic gradient approximations. We calculate all the m rewards for each (z, x, y) triplet and optimize the soft Q-learning loss (Guo et al., 2022). More details in Appendix §A.1.

4 Experiments

4.1 Tasks & Datasets

Unsupervised Text Style Transfer. We explore style transfer (Xu et al., 2012; Jin et al., 2022) into Shakespearean style. We consider three competing objectives: maintaining the original content of the input text, infusing it with Shakespearean style, and ensuring the resulting text conveys a positive sentiment. We test on the Shakespeare dataset (Xu et al., 2012; Jhamtani et al., 2017), and the reward function corresponding to content preservation is BertScore (Zhang et al., 2020), for sentiment is a sentiment RoBERTa-base classifier [2], and for style is a DistilBERT-base-uncased model fine-tuned on Shakespearean data. [3]

Method	Obj 1	Obj 2	Obj 3	Product	Average
Text Style Transfer (Obj_1: Content - Obj_2: Style Obj_3: Sentiment)					
Average	19.56	**79.25**	38.28	30.91	**45.69**
Product	**34.58**	57.78	35.11	**36.04**	42.49
HVI	25.39	67.91	**38.76**	32.44	44.02
MGDA	22.37	66.51	38.11	31.16	42.33
Machine Translation (Obj_1: Content - Obj_2: BLEU Obj_3: Sentiment)					
Average	32.07	**32.00**	46.36	65.48	36.81
Product	**32.95**	31.70	46.47	**65.98**	**37.04**
HVI	31.18	30.51	**48.69**	63.21	36.79
MGDA	31.46	31.85	46.03	62.87	36.45

Table 1: Reward values corresponding to each objective at a checkpoint where each method achieved the highest average of the product of rewards across samples. Even though the method utilizing the average of rewards achieved the highest average value for style transfer, we can observe an imbalance across various objective values. The product method, on the other hand, got highest product value, reflecting a more balanced improvement.

Supervised Machine Translation. We experiment on German to English translation task, using the *iwslt2017* data (Cettolo et al., 2017). The objectives here include: (1) semantic similarity between the generated translation and a reference text using BertScore, (2) BLEU score (Papineni et al., 2002) between generated text and reference, and (3) conveying a positive sentiment.

4.2 Training Details

Following (Deng et al., 2022), we consider a multi-layer perceptron head on top of a small frozen

[2] cardiffnlp/twitter-roberta-base-sentiment-latest
[3] notaphoenix/shakespeare_classifier_model

distilGPT-2 model (Sanh et al., 2019) as the policy network. The policy network is trained for 12,000 steps. The number of training samples used for text style transfer and machine translation are 100 and 200, respectively. At each step, we sample eight prompts for a given input from the policy network, each comprising five tokens. Subsequently, we feed each prompt along with its corresponding input text into a separate LM to generate 128 output samples. We use GPT-2 (Radford et al., 2019) for text style transfer and flan-T5-small (Chung et al., 2022) for machine translation tasks. We repeat each experiment with three distinct random seeds and report the average results. Using NVIDIA RTX A6000, each experiment takes about 20-24 hours.

5 Results

We compare the effectiveness of 4 methods - (1) average of rewards (Deng et al., 2022), (2) expected product of rewards, (3) HVI, and (4) MGDA. For the Shakespearean text-style transfer task, we show the pairwise objective values in Figure 2. Each data point on the scatter plot represents the average objective value computed from 128 output samples, where each sample is generated from a prompt sampled from the policy network and an input sentence from the validation dataset. Figure 2 illustrates how relying on the average of reward values can result in the sacrifice of individual objectives in favor of overall improvement. We observe instances where sentiment and style scores are notably low, despite a high content score. This phenomenon arises due to the emphasis placed solely on the average of rewards, without consideration for individual objectives. MGDA performs slightly better than the average reward when balancing the individual objectives. However, the HVI and the product of rewards improve all the objectives simultaneously, with greater success.

Similarly, we present the pairwise objective values for the machine translation task in figure 3 in Appendix §A.3. Again, we observe *objective collapse* for the average reward setting, while the other three approaches demonstrate a better balance among objectives while enhancing the joint reward. Notably, the hyper-volume approach and the product of rewards are more successful in optimizing all the objectives, simultaneously.

As the expected product of rewards serves as a reliable approximation of performance across objectives, we employ this metric to select a checkpoint for each approach. Subsequently, we report individual objective values, as well as their product and average across samples, in Table 1. When evaluating based on the product as the evaluation metric, the product method demonstrates superior performance compared to the other approaches. Additionally, we observe a more balanced improvement across all rewards with volume-based methods such as HVI and product, in contrast to Average and MGDA. For example, in style transfer task, the "average" method improves *style* disproportionately higher than other objectives despite achieving the best performance based on the Average metric.

6 Related Work

Prompt Tuning. A line of research has emerged with a focus on improving the discrete (Jiang et al., 2020; Prasad et al., 2023; Mishra et al., 2022) and soft prompts (Li and Liang, 2021; Qin and Eisner, 2021; Vu et al., 2022; Liu et al., 2023) for improved downstream performance. Few recent works generate discrete prompts by utilizing the models gradients (Shin et al., 2020; Wen et al., 2023), employing evolution algorithms (Guo et al., 2023), and reinforcement learning (Zhang et al., 2023; Deng et al., 2022; Jung and Kim, 2023; Wang et al., 2023). Our work shares a similar direction, but we focus on multiple competing objectives instead of one.

Multi-objective Reinforcement Learning. Multi-objective reinforcement learning is typically studied in decision-making (Van Moffaert et al., 2013; Van Moffaert and Nowé, 2014; Yang et al., 2019; Xu et al., 2020; Hayes et al., 2022). Jang et al. (2023) fine-tunes LMs for multiple objectives by training one policy model per objective and merging them. (Lin et al., 2019; Sener and Koltun, 2019) perform multi-objective RL in a multi-task learning setup. Instead, we propose optimizing the prompts for one model with multiple objectives.

7 Conclusion

We empirically investigate the use of optimization techniques alongside reinforcement learning to address discrete prompt optimization in a multi-objective context. Our experiments show that multi-objective methods, which directly optimize the volume, outperform those seeking monotonic update directions, achieving a better balance across all rewards.

8 Limitations

The methods discussed in this paper take many GPU hours to converge, making it computationally expensive to run. Moreover, our optimization methods perform well on smaller LMs like GPT2, we have not experimented with larger models.

9 Ethical Considerations

This paper introduces three approaches for discrete prompt optimization. As such, prompt-tuning should not introduce biases not already observed in the model and generate any harmful text as prompts, and we do not anticipate any significant ethical concerns.

References

Mauro Cettolo, Marcello Federico, Luisa Bentivogli, Jan Niehues, Sebastian Stüker, Katsuhito Sudoh, Koichiro Yoshino, and Christian Federmann. 2017. Overview of the IWSLT 2017 evaluation campaign. In *Proceedings of the 14th International Conference on Spoken Language Translation*, pages 2–14, Tokyo, Japan. International Workshop on Spoken Language Translation.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

Mingkai Deng, Jianyu Wang, Cheng-Ping Hsieh, Yihan Wang, Han Guo, Tianmin Shu, Meng Song, Eric Xing, and Zhiting Hu. 2022. RLPrompt: Optimizing discrete text prompts with reinforcement learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3369–3391, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jörg Fliege and Benar Fux Svaiter. 2000. Steepest descent methods for multicriteria optimization. *Mathematical methods of operations research*, 51:479–494.

Han Guo, Bowen Tan, Zhengzhong Liu, Eric P. Xing, and Zhiting Hu. 2022. Efficient (soft) q-learning for text generation with limited good data.

Qingyan Guo, Rui Wang, Junliang Guo, Bei Li, Kaitao Song, Xu Tan, Guoqing Liu, Jiang Bian, and Yujiu Yang. 2023. Connecting large language models with evolutionary algorithms yields powerful prompt optimizers. *arXiv preprint arXiv:2309.08532*.

Conor F Hayes, Roxana Rădulescu, Eugenio Bargiacchi, Johan Källström, Matthew Macfarlane, Mathieu Reymond, Timothy Verstraeten, Luisa M Zintgraf, Richard Dazeley, Fredrik Heintz, et al. 2022. A practical guide to multi-objective reinforcement learning and planning. *Autonomous Agents and Multi-Agent Systems*, 36(1):26.

Joel Jang, Seungone Kim, Bill Yuchen Lin, Yizhong Wang, Jack Hessel, Luke Zettlemoyer, Hannaneh Hajishirzi, Yejin Choi, and Prithviraj Ammanabrolu. 2023. Personalized soups: Personalized large language model alignment via post-hoc parameter merging. *arXiv preprint arXiv:2310.11564*.

Harsh Jhamtani, Varun Gangal, Eduard Hovy, and Eric Nyberg. 2017. Shakespearizing modern language using copy-enriched sequence to sequence models. In *Proceedings of the Workshop on Stylistic Variation*, pages 10–19, Copenhagen, Denmark. Association for Computational Linguistics.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Di Jin, Zhijing Jin, Zhiting Hu, Olga Vechtomova, and Rada Mihalcea. 2022. Deep learning for text style transfer: A survey. *Computational Linguistics*, 48(1):155–205.

Hoyoun Jung and Kyung-Joong Kim. 2023. Discrete prompt compression with reinforcement learning. *arXiv preprint arXiv:2308.08758*.

Joshua Knowles, David Corne, and Mark Fleischer. 2004. Bounded archiving using the lebesgue measure. volume 4, pages 2490 – 2497 Vol.4.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online. Association for Computational Linguistics.

Xi Lin, Hui-Ling Zhen, Zhenhua Li, Qing-Fu Zhang, and Sam Kwong. 2019. Pareto multi-task learning. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2023. Gpt understands, too. *AI Open*.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, Yejin Choi, and Hannaneh Hajishirzi. 2022. Reframing instructional prompts to GPTk's language. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 589–612, Dublin, Ireland. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. pages 311–318.

Archiki Prasad, Peter Hase, Xiang Zhou, and Mohit Bansal. 2023. GrIPS: Gradient-free, edit-based instruction search for prompting large language models. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3845–3864, Dubrovnik, Croatia. Association for Computational Linguistics.

Guanghui Qin and Jason Eisner. 2021. Learning how to ask: Querying LMs with mixtures of soft prompts. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5203–5212, Online. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In *NeurIPS EMC2 Workshop*.

Timo Schick and Hinrich Schütze. 2020. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Conference of the European Chapter of the Association for Computational Linguistics*.

Ozan Sener and Vladlen Koltun. 2019. Multi-task learning as multi-objective optimization.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4222–4235, Online. Association for Computational Linguistics.

Kristof Van Moffaert, Madalina M Drugan, and Ann Nowé. 2013. Scalarized multi-objective reinforcement learning: Novel design techniques. In *2013 IEEE symposium on adaptive dynamic programming and reinforcement learning (ADPRL)*, pages 191–199. IEEE.

Kristof Van Moffaert and Ann Nowé. 2014. Multi-objective reinforcement learning using sets of pareto dominating policies. *The Journal of Machine Learning Research*, 15(1):3483–3512.

Tu Vu, Brian Lester, Noah Constant, Rami Al-Rfou', and Daniel Cer. 2022. SPoT: Better frozen model adaptation through soft prompt transfer. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5039–5059, Dublin, Ireland. Association for Computational Linguistics.

Xinyuan Wang, Chenxi Li, Zhen Wang, Fan Bai, Haotian Luo, Jiayou Zhang, Nebojsa Jojic, Eric P Xing, and Zhiting Hu. 2023. Promptagent: Strategic planning with language models enables expert-level prompt optimization. *arXiv preprint arXiv:2310.16427*.

Yuxin Wen, Neel Jain, John Kirchenbauer, Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2023. Hard prompts made easy: Gradient-based discrete optimization for prompt tuning and discovery. *arXiv preprint arXiv:2302.03668*.

Jie Xu, Yunsheng Tian, Pingchuan Ma, Daniela Rus, Shinjiro Sueda, and Wojciech Matusik. 2020. Prediction-guided multi-objective reinforcement learning for continuous robot control. In *International conference on machine learning*, pages 10607–10616. PMLR.

Wei Xu, Alan Ritter, Bill Dolan, Ralph Grishman, and Colin Cherry. 2012. Paraphrasing for style. In *Proceedings of COLING 2012*, pages 2899–2914, Mumbai, India. The COLING 2012 Organizing Committee.

Runzhe Yang, Xingyuan Sun, and Karthik Narasimhan. 2019. A generalized algorithm for multi-objective reinforcement learning and policy adaptation. *Advances in neural information processing systems*, 32.

Tianjun Zhang, Xuezhi Wang, Denny Zhou, Dale Schuurmans, and Joseph E. Gonzalez. 2023. TEMPERA: Test-time prompt editing via reinforcement learning. In *The Eleventh International Conference on Learning Representations*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert.

Eckart Zitzler and Lothar Thiele. 1998. Multiobjective optimization using evolutionary algorithms — a comparative case study. In *Parallel Problem Solving from Nature — PPSN V*, pages 292–301, Berlin, Heidelberg. Springer Berlin Heidelberg.

A Appendix

A.1 Multiple Gradient Descent Algorithm

(Fliege and Svaiter, 2000) proposes a steepest descent algorithm for multi-criteria optimization, where the update rule for the parameters θ at time t with the step size η is defined as:

$$\theta_{t+1} = \theta_t - \eta d_t \tag{3}$$

where the search direction d_t is calculated as follows, with $\mathcal{L}_i(\theta_j)$ being the expected loss corresponding to objective o_i:

$$\begin{aligned} &(d_t, \alpha_t) = \arg \min_{d \in \mathbb{R}^n, \alpha \in \mathbb{R}} \alpha + \frac{1}{2}\|d\|^2, \\ &\text{s.t. } \nabla \mathcal{L}_i\left(\theta_t\right)^T d \leq \alpha, \quad i = 1, \ldots, m. \end{aligned} \tag{4}$$

A valid direction d_t improves the values for all the objectives, simultaneously. Moreover, (Fliege and Svaiter, 2000) shows that the solution obtained by the aforementioned approach leads to a Pareto critical point.

Based on the KKT conditions, we have

$$d_t = -\left(\sum_{i=1}^{m} \lambda_i \nabla \mathcal{L}_i\left(\theta_t\right)\right), \quad \sum_{i=1}^{m} \lambda_i = 1 \tag{5}$$

and we can write equation-4 in its dual form:

$$\begin{aligned} &\max_{\lambda_i} -\frac{1}{2}\left\|\sum_{i=1}^{m} \lambda_i \nabla \mathcal{L}_i\left(\theta_t\right)\right\|^2 \\ &\text{s.t. } \sum_{i=1}^{m} \lambda_i = 1, \lambda_i \geq 0, , \forall i = 1, \ldots, m. \end{aligned} \tag{6}$$

A.2 MGDA-based efficient Soft Q-learning for policy update

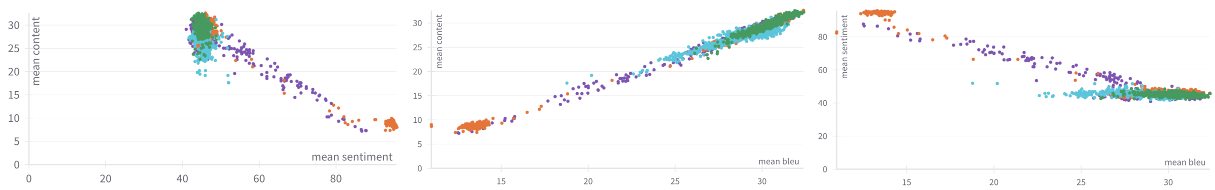
Algorithm 1 shows the procedure for policy updates using MGDA and soft Q-learning.

A.3 Pairwise reward values for Machine Translation Task

We present pairwise reward values for machine translation task for each method in Figure 3.

Algorithm 1 MGDA-based policy update for one input sentence

Input: Input sentence x, policy π_θ, reward models $r_{1...m}$, external frozen LM
$\{z_{1...k}\} \sim \pi_\theta(x)$ ▷ Sample k prompts from the policy
for i = 1 . . . k **do**:
    $\{y_{1...\hat{k}}\} \sim p_{LM}(y|x, z_i)$ ▷ Sample $\hat{k}$ output sentences from a desired LM
end for
for i = 1 . . . $k \cdot \hat{k}$ **do**:
    calculate $r_{1...m}(y_i, x)$ ▷ Calculate $r_{1...m}$ for each output sentence y and input x
end for
for i = 1 . . . m **do**:
    Calculate $\mathcal{L}_m$ using r_m ▷ Use efficient SQL loss (Guo et al., 2022)
end for
$\lambda_1, \ldots, \lambda_m = \text{FrankWolfeSolver}(\nabla_\theta \mathcal{L}_i(\theta))$ ▷ Find the direction using [6]
$\theta = \theta - \eta \sum_{i=1}^{m} \lambda_i \nabla_\theta \mathcal{L}_i(\theta)$ ▷ Gradient descent on policy parameters



Sentiment vs Content   BLEU vs Content   BLEU vs Sentiment

Figure 3: Pairwise reward values for Machine Translation Task from German to English, in different settings of average reward, hyper volume indicator reward, product reward, and multiple gradient descent algorithm.