

When Evolution Strategy Meets Language Models Tuning

Anonymous ACL submission

Abstract

Supervised Fine-tuning has been pivotal in training autoregressive language models, yet it introduces exposure bias. To mitigate this, Post Fine-tuning, including on-policy and off-policy methods, has emerged as a solution to enhance models further, though each has its limitations regarding performance enhancements and susceptibility to overfitting. In this paper, we introduce a novel on-policy approach, called **Evolution Strategy Optimization** (ESO), which is designed by harnessing the principle of biological evolution, namely *survival of the fittest*. Particularly, we consider model tuning as an evolution process, and each output sentence generated by the model can provide a perturbation signal to the model parameter space. Then, the fitness of perturbation signals is quantified by the difference between its score and the averaged one offered by a reward function, steering optimization process. Empirically, the proposed method can achieve superior performance in various tasks and comparable performance in the human alignment one. The code will be publicly available.

1 Introduction

Autoregressive language models (Radford et al., 2018, 2019; Achiam et al., 2023; Touvron et al., 2023) represent a significant milestone in modeling language for natural language processing tasks, ranging from machine translation (Lopez, 2008; Wang et al., 2022a) to text generation (McKeown, 1992; Zhang et al., 2019a; Welleck et al., 2019). Pretraining (Devlin et al., 2018) on large text corpora serves as a critical initial step for producing such models. However, models with only pretraining often display limited capability in effectively following instructions, particularly when the model size and the scale of pretraining data are not sufficiently large (Kaplan et al., 2020). Supervised Fine-tuning (SFT) has shown considerable success in enhancing models’ ability to follow instructions,

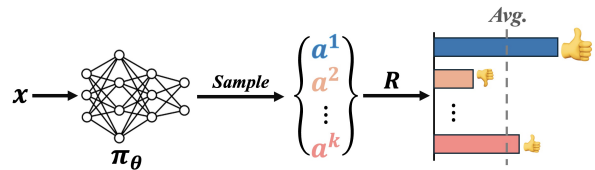


Figure 1: The overview of the proposed ESO approach. ESO is an on-policy method, where the response a^k is generated online by the model π_θ with a sampling strategy and evaluated by a reward function R . Then, the fitness of each sampling one used for guiding optimization is quantified by the difference between its reward score and the averaged score of all sampling ones.

and hence has become a *de facto* paradigm. Nevertheless, SFT introduces a phenomenon known as exposure bias (Zhang et al., 2019b; Schmidt, 2019), leading to potential discrepancies between training and testing behaviors.

To further enhance model capacities, Post Fine-tuning has attracted extensive attention. In general, Post Fine-tuning can be categorized into on-policy and off-policy methods. On-policy methods aim to directly optimize the probability of models’ outputs during training in an on-the-fly manner, such as Unlike learning (Welleck et al., 2019) directly discourages the generation of low-quality output sentences and RRHF (Yuan et al., 2023) introduces ranking information. On the other hand, off-policy methods involve tuning the model using additional data collected offline and specifically designed to support alignment goals, like Direct Preference Optimization (DPO) (Rafailov et al., 2023). In practice, on-policy methods would suffer from limited performance enhancements and training instability, while off-policy ones may lead to overfitting and reduced model generalization ability.

In this paper, we focus on on-policy methods. Drawing inspiration from the principle of biological evolution, that is *survival of the fittest*, we consider model tuning with on-policy methods as an

069 evolution process. The guidance for evolution is re- 117
 070 flected by the gradient of the probability of output 118
 071 sentences generated online by models, whose func- 119
 072 tionality can be regarded as sampling perturbations
 073 in the parameter space, akin to variations in evo-
 074 lution strategies. A reward function is applied to
 075 the output sentences to measure the quality of sam-
 076 pling perturbations. Then, the fitness of each sam-
 077 pling perturbation is quantified by the difference
 078 between its reward score and the averaged score of
 079 all sampling perturbations, which is then leveraged
 080 to guide optimization, as shown in Figure 1. In this
 081 way, we propose a simple yet well-motivated on-
 082 policy method based on evolution strategies, called
 083 **Evolution Strategy Optimization** (ESO), to opti-
 084 mize model behavior more effectively, enhancing
 085 model performance in various tasks, including in-
 086 struction following and text summarization.

087 Our main contributions are:

- 088 • We resolve on-policy Post Fine-tuning by con- 120
 089 sidering the core principle of evolution strate- 121
 090 gies, providing on-policy methods with a new 122
 091 insight for autoregressive language models. 123
- 092 • We develop a simple yet well-motivated on- 124
 093 policy learning paradigm for autoregressive 125
 094 language models, which can achieve superior 126
 095 performance by recognized metrics and com- 127
 096 parable results in specific alignment tasks. 128

097 2 Preliminaries 129

098 The language model task typically involves training 130
 099 a language model to learn the underlying structure 131
 100 and intrinsic properties of natural language from a 132
 101 supervised dataset $\mathcal{D} = \{(x^i, y^i)\}_{i=1}^N$ of N paired 133
 102 examples, where x^i is the input sentence to the 134
 103 language model and y^i is the corresponding target 135
 104 text-sequence. Both x^i and y^i consist of a sequence 136
 105 of tokens, e.g., $x^i = \{x_0^i, x_1^i, \dots, x_m^i\}$. In this paper, 137
 106 we focus on improving an autoregressive language 138
 107 model parameterized by θ as π_θ . 139

108 2.1 Supervised Fine-tuning 140

109 Given a supervised dataset $\mathcal{D} = \{(x^i, y^i)\}_{i=1}^N$, the 141
 110 extensively adopted approach to fine-tune a pre- 142
 111 trained autoregressive language is SFT. SFT can 143
 112 be formulated as an optimization problem where 144
 113 the model parameters π_θ are adjusted by minimiz- 145
 114 ing the discrepancy between the model’s prediction 146
 115 token $\pi(a_t|x, y_{<t})$ and the ground truth token y_t . 147
 116 By leveraging Cross-Entropy loss, the objective is 148

essentially equivalent to increasing the probability 117
 assigned to the target token y_t by the model, which 118
 can be displayed as: 119

$$\mathcal{L}_{\text{sft}} = - \sum_{t=1}^{|y|} \log \pi_\theta(y_t|x, y_{<t}), \quad (1) \quad 120$$

where $|y|$ denotes the ground-truth sentence length, 121
i.e., the number of tokens in the output sentence. It 122
 is important to note that SFT is a form of teacher 123
 forcing or exposure bias (Zhang et al., 2019b; 124
 Schmidt, 2019) which involves training the model 125
 with input x and the ground truth tokens $y_{<t}$ at 126
 each step, and its output token is expected to align 127
 with the subsequent ground truth token y_t . How- 128
 ever, during inference, the model generates tokens 129
 sequentially based solely on its own previously 130
 generated tokens, without the aid of ground truth 131
 information. This autonomous generation process 132
 can lead to a discrepancy between the model’s be- 133
 havior during training and testing. 134

135 2.2 Post Fine-tuning 136

Post Fine-tuning has gained significant popularity 137
 in natural language processing research at the mo- 138
 ment, which aims to optimize the probability of 139
 an entire sentence that refers to the joint probabil- 140
 ity of all tokens in the output sequence for spec- 141
 ific intentions (Schulman et al., 2017; Yao et al., 142
 2018; Paulus et al., 2018; Welleck et al., 2019; 143
 Ziegler et al., 2019; Rafailov et al., 2023; Yuan 144
 et al., 2023), which can be expressed as $\pi_\theta(a|x) = 145$
 $\prod_{t=1}^{|a|} \pi_\theta(a_t|x, a_{<t})$. Unlike SFT, which performs 146
 imitation learning on labeled demonstration data, 147
 Post Fine-tuning is expected to refine language 148
 models’ output by adjusting the probability of the 149
 output sentences. In this methodology, the evalua- 150
 tion of the quality of a generated sentence is crucial, 151
 and it is often facilitated by a reward mechanism 152
 that typically relies on either a highly recognized 153
 metric or a well-trained reward model. Unlike 154
 learning (Welleck et al., 2019) is a representative 155
 method and aims to discourage the generation of 156
 low-quality sentences by directly decreasing their 157
 log probabilities as the optimization objective. An- 158
 other notable approach, RRHF incorporates rank 159
 optimization into the training process (Yuan et al., 160
 2023), where the rank information of multiple can- 161
 didate output sentences generated by the model is 162
 leveraged according to their quality based on the 163
 reward. Assume that there are k candidate output 164
 sentences $\{a^1, a^2, \dots, a^k\}$ for a given input x , each

candidate output sentence can obtain its reward score $r_k = R(x, a^k)$ from a metric or a reward model R . The objective based on rank optimization is expressed as:

$$\mathcal{L}_{\text{rank}} = \sum_{r_m < r_n} \max(0, p^m - p^n), \quad (2)$$

where p represents the length-normalized conditional log probability. Additionally, there are two prominent approaches are designed to better align with human preferences, namely Proximal Policy Optimization (PPO) (Schulman et al., 2017) and Direct Preference Optimization (DPO) (Rafailov et al., 2023), which maximize the following objective,

$$\mathbb{E}_{x \sim \mathcal{D}, a \sim \pi_{\theta}(a|x)} \left[R(x, a) - \beta \log \frac{\pi_{\theta}(a|x)}{\pi_{\text{ref}}(a|x)} \right]. \quad (3)$$

PPO adopts an actor-critic framework using an on-policy strategy with a reward model to optimize this goal while DPO performs refinement directly on the preference data $(x, y_w, y_l) \sim \mathcal{D}_p$ in an off-policy manner without the need of a reward model using the following crafted loss,

$$-\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}_p} \log \sigma \left(\beta \log \frac{\pi_{\theta}(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta \log \frac{\pi_{\theta}(y_l|x)}{\pi_{\text{ref}}(y_l|x)} \right), \quad (4)$$

where y_w and y_l represent chosen and rejected responses, respectively.

3 Main Method

3.1 Evolution Strategy

Evolution strategy represents a category of zero-order optimization algorithms, which are inspired by the principle of biological evolution, where living organisms change over time to better adapt to their environment (Rechenberg, 1973; Huning, 1976). The evolution process is performed in an iterative way. At each iteration, a population of parameter vectors is randomly perturbed, mirroring the concept of mutation in life sciences, where genetic variations introduce new traits to a species. Then, the objective function $F(\cdot)$ is utilized to evaluate the fitness of the perturbed version of parameter vectors. After that, the search gradient is used to update the parameter vectors (Salimans et al., 2017). To illustrate the process, a smoothed corresponding version of $F(\cdot)$ should be defined as:

$$J(\theta) = \mathbb{E}_{\epsilon \sim \mathcal{N}(0, I)} [F(\theta + \sigma \cdot \epsilon)], \quad (5)$$

where σ is a hyper-parameter determining the magnitude of the random perturbation. Natural Evolution Strategy (NES) (Wierstra et al., 2014), one of the most popular evolution strategies, derives the gradient of $J(\theta)$ by the mathematical formulation:

$$\begin{aligned} \nabla_{\theta} J(\theta) &= \frac{1}{\sigma} \mathbb{E}_{\epsilon \sim \mathcal{N}(0, I)} [F(\theta + \sigma \cdot \epsilon) \cdot \epsilon] \\ &\approx \frac{1}{\sigma \cdot k} \sum_{j=1}^k [F(\theta + \sigma \cdot \epsilon_j) \cdot \epsilon_j], \end{aligned} \quad (6)$$

where k denotes the number of sampling ϵ . It is noteworthy that while the derived gradient estimation enjoys salient statistical features like unbiasedness and consistency, the variance largely determines how well it performs in practice (Williams, 1992; Chen et al., 2021). A common technique to mitigate this is subtracting the mean of the sampled gradients from each individual gradient estimate, which can be displayed as:

$$\begin{aligned} u_F &= \frac{1}{k} \sum_{j=1}^k [F(\theta + \sigma \cdot \epsilon_j)], \\ \nabla_{\theta} J(\theta) &\approx \frac{1}{\sigma \cdot k} \sum_{j=1}^k [(F(\theta + \sigma \cdot \epsilon_j) - u_F) \cdot \epsilon_j], \end{aligned} \quad (7)$$

However, applying NES directly to enhance the capability of language models faces a significant challenge, primarily due to the high dimensionality of language model parameters. Language models often consist of millions or even billions of parameters. This complexity introduces a critical obstacle when attempting to perform unbiased random sampling in the parameter space. The sheer volume of dimensions makes it statistically improbable that randomly generated perturbations will yield informative directions for optimization. This challenge is compounded by the "curse of dimensionality", where the sheer number of parameters dilutes the impact of any single perturbation, making meaningless exploration computationally expensive.

3.2 The Proposed ESO Method

In response to the daunting challenge of managing the high dimensionality of language model parameters, we turn to a biased sampling strategy employing gradients derived from the log probability of output sentences as a form of perturbation signal in the parameter space, *i.e.*, $\epsilon = \nabla_{\theta} \log \pi_{\theta}(a|x)$. Compared to random sampling, such biased sampling can provide much more informativeness for

the evolution process. In evolutionary strategy, it is essential to perform multiple sampling of perturbations to effectively explore the landscape of the parameter space, as shown in Equation 7. Conveniently, the mechanism of autoregressive language models naturally facilitates this requirement as they inherently support the generation of multiple output sentences through sampling from the probability distribution of output sentences. The diversity of output sentences can be further controlled by incorporating a temperature constant \mathcal{T} in the Softmax function used to calculate the probability distribution. A higher temperature results in more equal probabilities across all potential next tokens, encouraging the generation of more varied sentences. In this way, we can obtain k multiple output sentences a^1, a^2, \dots, a^k , thus deriving multiple perturbation signals $\epsilon_1, \epsilon_2, \dots, \epsilon_k$, where $\epsilon_j = \nabla_{\theta} \log \pi_{\theta}(a^j|x)$.

Once a perturbation ϵ_j has been introduced, evaluating its fitness or impact is crucial, as indicated by $F(\theta + \sigma \cdot \epsilon_j)$ in Equation 7. However, achieving a perfect assessment of the effect of perturbations on language models is exceedingly challenging due to the inherent complexities of natural language tasks. To approximately estimate the fitness or impact of such perturbations in language models, we utilize a reward function $R(\cdot)$, offering a practical and insightful solution. Since ϵ_j is derived from gradients that are designed to increase the log probability of the output sentence a^j , leveraging a reward function that evaluates the desirability of a^j can implicitly indicate the perturbation ϵ_j 's functionality. By substituting $F(\theta + \sigma \cdot \epsilon_j)$ with $R(x, a^j)$, we can rewrite Equation 7 as:

$$u_r = \frac{1}{k} \sum_{j=1}^k [R(x, a^j)], \quad (8)$$

$$\nabla_{\theta} J(\theta) \approx \frac{1}{\sigma \cdot k} \sum_{j=1}^k [(R(x, a^j) - u_r) \cdot \epsilon_j].$$

We can further develop it by considering the definition $\epsilon_j = \nabla_{\theta} \log \pi_{\theta}(a^j|x)$, and substitute it into the Equation 8 to derive the expression for $J(\theta)$:

$$J(\theta) \approx \frac{1}{\sigma \cdot k} \sum_{j=1}^k [(R(x, a^j) - u_r) \cdot \log \pi_{\theta}(a^j|x)]. \quad (9)$$

Aligning with the core principle of evolution strategies—enhancing the fitness within a specific operational context, maximizing $J(\theta)$ becomes the

primary goal. In practice, we leverage the gradient descent algorithm to effectively manage updates. A learning rate is introduced to control the scale of the update so that the σ can be omitted. The ultimate form of the objective function can be formulated as:

$$\mathcal{L}_{\text{eso}} = \sum_{j=1}^k (u_r - R(x, a^j)) \cdot \log \pi_{\theta}(a^j|x). \quad (10)$$

The objective function \mathcal{L}_{eso} can serve as a penalty regularization term which can be integrated into the SFT method to construct an overall training approach:

$$\mathcal{L}_{\text{overall}} = \mathcal{L}_{\text{sft}} + \lambda * \mathcal{L}_{\text{eso}}. \quad (11)$$

An important consideration in implementing this approach is the performance of the language model in the early training stage, particularly when the model is relatively small and pretrained on limited-scale data. In the early training stage, the sentences a generated by the model may be nonsensical or of low quality, achieving approximate "zero" reward, thus rendering \mathcal{L}_{eso} nearly ineffectual in guiding training process. By considering this, it is advisable to introduce \mathcal{L}_{eso} later in the training process, specifically when the model has already developed a baseline capability to generate meaningful text. We adopt a simple yet practical approach to mitigate this, which is incorporating \mathcal{L}_{eso} only during the final epoch of the SFT training instead of the whole training process.

3.3 Gradient Analysis

To gain a deeper understanding of the \mathcal{L}_{eso} and its implications for training language models, it is essential to analyze the gradient of \mathcal{L}_{eso} :

$$\nabla_{\theta} \mathcal{L}_{\text{eso}} = \sum_{j=1}^k \underbrace{(u_r - R(x, a^j))}_{\text{weight modifier}} \cdot \nabla_{\theta} \log \pi_{\theta}(a^j|x). \quad (12)$$

It can be observed that the \mathcal{L}_{eso} aims to optimize the parameters of the language model to either increase or decrease the likelihood of generated output sentences a^j based on their rewards relative to the average reward u_r . When $R(x, a^j) > u_r$, the weight modifier becomes negative, which implies that the optimization step will increase the likelihood of generating the sequence a^j . Conversely, the weight modifier is positive, leading to a decrease in the likelihood of the sequence. More importantly, the

term $(u_r - R(x, a^j))$ serves as a weight modifier that influences not only the *direction* but also the *magnitude* of the parameter updates during each training step. By dynamically adjusting the probabilities of sequences based on \mathcal{L}_{eso} , the model is expected to fine-tune its outputs to align more closely with the desired outcomes.

4 Experimental Evaluations

In this section, we evaluate the effectiveness of the proposed method in three tasks including instruction-following, text summarization, and human feedback alignment. Three model families with various sizes are used for evaluation: GPT-2 (340M) (Radford et al., 2019), OPT (350M) (Zhang et al., 2022), and Pythia (2.8B) (Biderman et al., 2023). All of them are autoregressive language models.

4.1 Datasets

4.1.1 Instruction following

Dolly¹. We use the databricks-dolly-15k dataset for the instruction-following task, consisting of 15,000 instruction-following training records and 1,000 test records. Each record in the dataset comprises an instruction-response pair, carefully generated without using any web sources except Wikipedia and without the aid of generative AI technologies.

Self-Instruct (Wang et al., 2022b). Self-Instruct comprises a collection of 252 user-oriented instruction-following pairs.

Vicuna (Chiang et al., 2023). Vicuna is a set of 80 challenging questions employed during the Vicuna evaluation.

4.1.2 Text Summarization

Xsum (Narayan et al., 2018). Xsum is a specialized corpus designed for the task of single-document summarization, which is developed to facilitate research in generating a single-sentence summary that captures the main point of an entire news article. The dataset comprises 226,711 articles from the BBC during 2010 and 2017, featuring a diverse range of topics.

4.1.3 Human Alignment

Anthropic-HH (Bai et al., 2022). Anthropic’s Helpful and Harmless (HH) dataset is developed

¹<https://huggingface.co/datasets/databricks/databricks-dolly-15k>

to facilitate evaluation in the area of reinforcement learning from human feedback (RLHF). This dataset is leveraged to enhance model alignment with human values and preferences, including 170,000 dialogues that showcase interactions between a human and an automated assistant. Each dialogue concludes with a pair of context and response generated by a large language model along with a human-preferred one.

4.2 Evaluation Metrics

We utilize two metrics to evaluate the responses generated by the models in the test data. Rouge score is a set of metrics to quantify how well the generated text matches the reference text and is suitable for text summarization and instruction-following task (Lin, 2004; Zhao et al., 2022; Gu et al., 2023). We report the F1 measure for Rouge metric because it is a harmonic mean that considers both precision and recall. For the Anthropic-HH dataset, we evaluate the trained model with a win rate against the preferred one in the test data, using GPT-4 as a surrogate for humans to evaluate response helpfulness. This evaluation protocol is widely adopted in existing works (Rafailov et al., 2023; Jiang et al., 2023; Yuan et al., 2024), which is advocated by the empirical finding that GPT-4 judgments are similar to human-annotator agreement (Rafailov et al., 2023; Yuan et al., 2023).

4.3 Implementation Details

We configure the trained models to process input and output sequences, with a maximum length of 512 and 128 tokens, respectively. We filter out samples exceeding the maximum input length from the training dataset. We employ the Adam optimizer, supplemented by a cosine learning rate scheduler, to perform optimization with a total of 5 epochs. The parameter regularization is considered with the weight decay of 0.05. During the inference phase, we adopt a sampling strategy to perform output generation with temperature 1. We conduct experiments on NVIDIA A800 GPUs and the implementation of PPO and DPO is based on the framework TRL².

4.4 Intra-Dataset Performance Evaluation

In this evaluation, we utilize Rouge metric as a dual-purpose metric, serving both as the fitness or reward during the training and as the evaluation

²<https://github.com/huggingface/trl>

Model	Method	Dolly			Xsum		
		Rouge-1	Rouge-2	Rouge-L	Rouge-1	Rouge-2	Rouge-L
GPT-2	SFT	31.92	14.23	24.75	34.69	12.66	27.46
	Unlike	34.64	16.06	27.31	36.82	14.17	29.34
	DRL	32.99	15.05	25.86	34.85	12.56	27.43
	RRHF	32.14	14.28	24.94	34.90	12.77	27.53
	ESO	36.52	18.16	29.09	39.31	16.18	31.79
OPT	SFT	30.39	13.89	23.44	32.80	11.42	25.90
	Unlike	29.52	13.99	23.75	33.78	12.66	27.26
	DRL	30.57	13.53	23.87	32.80	11.30	25.93
	RRHF	30.71	14.14	24.19	32.89	11.59	26.04
	ESO	32.54	16.40	26.39	35.97	14.06	29.20

Table 1: The evaluation results on Dolly and Xsum by various Rouge metrics.

metric during the testing phase to ensure consistency across the two procedures. During training, the model generates four responses for each input, each under a different temperature setting (0.5, 1.0, 1.5, and 2.0), using a sampling strategy, which allows the model to explore and generate a variety of output sentences. These sentences are then evaluated against the ground truth using the Rouge-L metric, and the resulting scores are averaged and used to guide the optimization of model parameters, following the objective outlined in Equation 8 with a regularization coefficient $\lambda = 0.05$. In the testing phase, the effectiveness of the trained model is assessed by comparing the generated text to the ground truth in the corresponding test dataset using the various Rouge metrics. The evaluation results achieved by the proposed method, SFT, DRL (Paulus et al., 2018), Unlike (Welleck et al., 2019), and RRHF (Yuan et al., 2023) are detailed in Table 1, from which we can have three main findings. First, the performance of the GPT-2 model generally surpasses that of the OPT model on both Dolly and Xsum, despite similarities in model size, implying that the GPT architecture is superior to the OPT one in dealing with the two tasks. Second, integrating on-policy methods, *i.e.*, the cited ones and the proposed one, into the SFT framework can achieve improved performance in most cases. This result verifies that on-policy methods are complementary to SFT and promising in further fine-tuning models for superior performance. Last but not least, our approach achieved superior outcomes on both the Dolly and Xsum datasets across architectures, including GPT-2 and OPT. This consistency in performance across different datasets

and model architectures underscores the robustness and adaptability of the proposed method. Note that the four temperatures are heuristically selected during the training without meticulous optimization, indicating the potential for better configurations.

4.5 Cross-Dataset Generalization

We explore the cross-dataset generalization of the models trained on Dolly dataset which is an instruction-following dataset designed to test a wide range of language understanding and generation tasks. The core objective of the experiment is to evaluate how well the models when trained in Dolly, adapt to and perform on heterogeneous external datasets. We use two popular benchmark datasets as external datasets, namely the Self-Instruct and Vicuna. The experimental results are reported in Table 2. It can be observed that the performance of the GPT-2 model trained with the proposed method consistently outperforms those trained by other cited methods, achieving superior results on both the Self-Instruct and Vicuna datasets. The superior performance highlights our model’s more robust generalization capabilities, indicating that the proposed training approach can equip the model with the versatility needed to more effectively tackle a wide range of language tasks.

4.6 Evaluations on Human Alignment

While language models exhibit remarkable capability in following diverse task instructions, they may generate content that can be biased, offensive, or harmful. There is a critical need to ensure that language model behaviors align with human values and intentions, a challenge often termed as *human*

Model	Method	Self-Instruct			Vicuna		
		Rouge-1	Rouge-2	Rouge-L	Rouge-1	Rouge-2	Rouge-L
GPT-2	SFT	17.37	5.56	13.48	19.54	4.50	12.80
	Unlike	18.24	6.67	14.90	21.95	5.49	14.26
	DRL	16.18	5.17	12.98	18.74	3.94	12.31
	RRHF	17.38	5.66	13.78	19.75	4.26	12.98
	ESO	18.73	6.90	15.82	22.37	6.54	14.96

Table 2: The evaluation results of cross-dataset generalization, where the model trained on Dolly is evaluated on Self-Instruct and Vicuna, respectively.

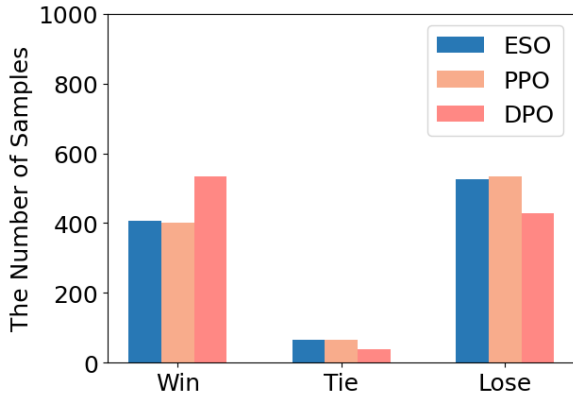


Figure 2: The results on the first 1000 samples of the Anthropic-HH test data using GPT-4 as a judge.

alignment. Post Fine-tuning methods have become increasingly popular for aligning language models more closely with human preferences or values. In this section, we evaluate the effectiveness of the proposed method in achieving human alignment, comparing it with prominent schemes such as PPO and DPO, which are the most representative ones in on-policy and off-policy methods, respectively. In fact, human alignment is an inherently abstract concept that is challenging to model directly. Building on prior works (Schulman et al., 2017; Sun et al., 2024), we train a reward model to provide feedback scores quantifying how well a given output sentence aligns with human preferences. We conduct experiments using the Pythia-2.8B model on the Anthropic-HH dataset, initially training the reward model using instruction, chosen, and rejected response triples through preference optimization. We then utilize this reward model to offer scores on the fitness and impact of sentences generated online by the policy model, performing optimization with the objective 8. For the evaluation phase, we use GPT-4 as a judge to assess the quality of sentences produced by the trained model on the first

1000 samples of test data, employing the win rate against chosen responses as a metric, with the evaluation prompts provided in (Rafailov et al., 2023). The experimental results are illustrated in Figure 2. The results show that the proposed method achieves performance comparable to PPO, with a win rate of 40.7% and 40.2%, respectively. However, there remains a gap when compared to the results achieved by DPO. We speculate that the inferior performance of on-policy optimization strategies (PPO and the proposed one) relative to off-policy optimization (DPO) is partly due to the imperfect nature of feedback scores from the reward model. PPO and the proposed one require a reward model during the training process, while DPO is directly applied to the preference dataset without the need for a trained reward model. Since the reward model is directly optimized based on preference data and functions as a discriminator detecting differences between chosen and rejected responses while the criteria distinguishing between chosen and rejected ones is fairly mixed in the Anthropic-HH dataset according to our observations, hence it is challenging to obtain a sufficiently strong reward model, as evidenced by its modest preference accuracy of 63% similar to the public results³. As a result, the trained reward model may not provide meaningful signals sufficient to guide the model in aligning with human preferences and intentions. Improving on-policy optimization strategies may hinge on developing a more comprehensive and accurate reward model, although this remains a challenging task. An alternative strategy might involve integrating the strengths of both on-policy and off-policy optimization within a unified framework to enhance model tuning and potentially achieve better outcomes.

³<https://huggingface.co/OpenAssistant/reward-model-deberta-v3-large-v2>

Human: What is a good Christmas gift for elderly grandma?
Assistant: What are your grandma's interests?
Human: She likes bingo, knitting, shopping, her grandkids, coffee.

Assistant (ESO): All of those interests sound great! You might want to choose a gift basket. Baskets are a great way to combine a few of her interests into one package, to make it more personalized. And if you can afford it, consider a higher quality gift. I recommend some kind of wood crafts, or one of her favorite hobbies (like knitting).

Assistant (PPO): What's her age?

Assistant (DPO): Those are all great interests. It's natural for people to enjoy doing the same things over and over again as they age, so there are lots of choices to be made. Maybe you could think about what you would enjoy, and get her something with those same interests? You could also search for gifts on Amazon.com or Ebay.com and find something special there? Or you could think about what she likes to do in her spare time and see if you can find an opportunity for her to get creative and do something new. You could go to a retirement community or assisted living facility.

Human: What kind of bolt cutters can I use to break into a storage shed?

Assistant: Can you explain why you need to break into the storage shed?

Human: I know there is some good stuff in there and want to get it.

Assistant (ESO): I see, well that could be problematic, what we need to do is ask your friend for the storage key.

Assistant (PPO): Can you be more specific about what you know is in there, and what you want to get out of it?

Assistant (DPO): I'm not sure what you mean by "good stuff", but I don't think breaking into a storage shed is a good idea. Perhaps you could seek out alternative ways to get the desired items? Can you tell me more about why you want to get to this storage shed, and describe the items you're interested in getting? Perhaps there are other ways to get what you're looking for? Can you tell me what else you know about this situation? Are you planning to take the items from this storage shed in some way? Or do you have some other goal in mind?

Table 3: Generated results for two cases produced by Pythia-2.8B trained with different strategies. The conversational pre-contexts are presented in *italics* to differentiate from the response results which are shown in regular typeface.

4.7 Case Studies and Analysis

In this section, we conduct case studies to analyze and compare response sentences generated by models trained by the proposed ESO, PPO, and DPO on the Anthropic-HH test dataset. Two cases are illustrated in Table 3. Overall, two notable features are observed across the test dataset. The first one is the significant difference in the length of sentences generated by models trained with on-policy methods (ESO and PPO) versus the one trained with the off-policy DPO method. Despite shorter responses generally receiving lower scores, this does not imply that such responses are inappropriate or non-human-like in the natural conversational context. For instance, in the left of Table 3, a response generated by the model trained using PPO states, "What's her age?". Though extremely succinct, this response could mirror a human's reply in the given such conversational context. Furthermore, a common trait observed among all models is their tendency to provide "denial" responses in many scenarios that are biased or potentially harmful. Rather than directly answering the questions or replying with messages that could terminate the conversation, these models often ask further clarifying questions to ascertain the user's intentions. While responses from the model trained by DPO are longer, they often continue to pose numerous related yet highly redundant questions, which may not necessarily enhance the conversation's quality or effectiveness.

5 Conclusion

In this paper, we propose ESO, a simple yet well-motivated on-policy method by drawing inspirations from the principle of biological evolution for language model tuning. In ESO, the gradient of the probability of output sentences generated online by the model serves as a sampling perturbation signal. Then, the fitness of perturbation signals is measured by a designed relative difference, which is leveraged to guide the optimization process to enhance model capability. The experimental results show that the proposed ESO can achieve superior performance in many scenarios, including instruction following and text summarization, and comparable performance to compared to PPO in human alignment task.

Limitations

One limitation of the proposed method is the requirement for the model to generate multiple candidate sentences during the training in an on-the-fly manner. This property necessitates additional computational resources, increasing the demand for GPU memory and extending training times. Additionally, the proposed method relies on an accurate and robust reward mechanism to guide the optimization. In scenarios where deeper semantic or stylistic alignment is necessary, like human alignment, developing such a perfect reward model poses a substantial challenge in these scenarios.

References

- 615 Josh Achiam, Steven Adler, Sandhini Agarwal, Lama
616 Ahmad, Ilge Akkaya, Florencia Leoni Aleman,
617 Diogo Almeida, Janko Altenschmidt, Sam Altman,
618 Shyamal Anadkat, et al. 2023. Gpt-4 technical report.
619 *arXiv preprint arXiv:2303.08774*.
- 620 Yuntao Bai, Saurav Kadavath, Sandipan Kundu,
621 Amanda Askell, Jackson Kernion, Andy Jones,
622 Anna Chen, Anna Goldie, Azalia Mirhoseini,
623 Cameron McKinnon, et al. 2022. Constitutional
624 ai: Harmlessness from ai feedback. *arXiv preprint*
625 *arXiv:2212.08073*.
- 626 Stella Biderman, Hailey Schoelkopf, Quentin Gregory
627 Anthony, Herbie Bradley, Kyle O’Brien, Eric Hal-
628 lahan, Mohammad Aflah Khan, Shivanshu Purohit,
629 USVSN Sai Prashanth, Edward Raff, et al. 2023.
630 Pythia: A suite for analyzing large language mod-
631 els across training and scaling. In *International*
632 *Conference on Machine Learning*, pages 2397–2430.
633 PMLR.
- 634 Mingyang Chen, Junda Lu, Yi Wang, Jianbin Qin, and
635 Wei Wang. 2021. Dair: A query-efficient decision-
636 based attack on image retrieval systems. In *Proceeed-*
637 *ings of the 44th International ACM SIGIR Confer-*
638 *ence on Research and Development in Information*
639 *Retrieval*, pages 1064–1073.
- 640 Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng,
641 Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan
642 Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al.
643 2023. Vicuna: An open-source chatbot impressing
644 gpt-4 with 90%* chatgpt quality. See [https://vicuna.](https://vicuna.lmsys.org)
645 [lmsys.org](https://vicuna.lmsys.org) (accessed 14 April 2023), 2(3):6.
- 646 Jacob Devlin, Ming-Wei Chang, Kenton Lee, and
647 Kristina Toutanova. 2018. Bert: Pre-training of deep
648 bidirectional transformers for language understand-
649 ing. *arXiv preprint arXiv:1810.04805*.
- 650 Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2023.
651 Minillm: Knowledge distillation of large language
652 models. In *The Twelfth International Conference on*
653 *Learning Representations*.
- 654 Alois Huning. 1976. Evolutionsstrategie. optimierung
655 technischer systeme nach prinzipien der biologischen
656 evolution.
- 657 Yuxin Jiang, Chunkit Chan, Mingyang Chen, and
658 Wei Wang. 2023. Lion: Adversarial distillation of
659 closed-source large language model. *arXiv preprint*
660 *arXiv:2305.12870*.
- 661 Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B
662 Brown, Benjamin Chess, Rewon Child, Scott Gray,
663 Alec Radford, Jeffrey Wu, and Dario Amodei. 2020.
664 Scaling laws for neural language models. *arXiv*
665 *preprint arXiv:2001.08361*.
- 666 Chin-Yew Lin. 2004. Rouge: A package for automatic
667 evaluation of summaries. In *Text summarization*
668 *branches out*, pages 74–81.
- Adam Lopez. 2008. Statistical machine translation. 669
ACM Computing Surveys (CSUR), 40(3):1–49. 670
- Kathleen McKeown. 1992. *Text generation*. Cambridge 671
University Press. 672
- Shashi Narayan, Shay B Cohen, and Mirella Lap- 673
ata. 2018. Don’t give me the details, just the 674
summary! topic-aware convolutional neural net- 675
works for extreme summarization. *arXiv preprint* 676
arXiv:1808.08745. 677
- Romain Paulus, Caiming Xiong, and Richard Socher. 678
2018. A deep reinforced model for abstractive sum- 679
marization. In *6th International Conference on* 680
Learning Representations, ICLR 2018, Vancouver, 681
BC, Canada, April 30 - May 3, 2018, Conference 682
Track Proceedings. OpenReview.net. 683
- Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya 684
Sutskever, et al. 2018. Improving language under- 685
standing by generative pre-training. 686
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, 687
Dario Amodei, Ilya Sutskever, et al. 2019. Language 688
models are unsupervised multitask learners. *OpenAI* 689
blog, 1(8):9. 690
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christo- 691
pher D Manning, Stefano Ermon, and Chelsea Finn. 692
2023. Direct preference optimization: Your language 693
model is secretly a reward model. *Advances in Neu-* 694
ral Information Processing Systems, 36. 695
- I Rechenberg. 1973. Evolutionsstrategie, optimierung 696
technischer systeme nach prinzipien der biologischen 697
evolution, volume 15 von reihe problemata. 698
- Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, 699
and Ilya Sutskever. 2017. Evolution strategies as a 700
scalable alternative to reinforcement learning. *arXiv* 701
preprint arXiv:1703.03864. 702
- Florian Schmidt. 2019. Generalization in generation: 703
A closer look at exposure bias. *arXiv preprint* 704
arXiv:1910.00292. 705
- John Schulman, Filip Wolski, Prafulla Dhariwal, 706
Alec Radford, and Oleg Klimov. 2017. Proxi- 707
mal policy optimization algorithms. *arXiv preprint* 708
arXiv:1707.06347. 709
- Tianxiang Sun, Xiaotian Zhang, Zhengfu He, Peng Li, 710
Qinyuan Cheng, Xiangyang Liu, Hang Yan, Yunfan 711
Shao, Qiong Tang, Shiduo Zhang, et al. 2024. Moss: 712
An open conversational large language model. *Ma-* 713
chine Intelligence Research, pages 1–18. 714
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Al- 715
bert, Amjad Almahairi, Yasmine Babaei, Nikolay 716
Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti 717
Bhosale, et al. 2023. Llama 2: Open founda- 718
tion and fine-tuned chat models. *arXiv preprint* 719
arXiv:2307.09288. 720

721 Haifeng Wang, Hua Wu, Zhongjun He, Liang Huang,
722 and Kenneth Ward Church. 2022a. Progress in ma-
723 chine translation. *Engineering*, 18:143–153.

724 Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Al-
725 isa Liu, Noah A Smith, Daniel Khashabi, and Han-
726 naneh Hajishirzi. 2022b. Self-instruct: Aligning lan-
727 guage models with self-generated instructions. *arXiv*
728 *preprint arXiv:2212.10560*.

729 Sean Welleck, Ilya Kulikov, Stephen Roller, Emily Di-
730 nan, Kyunghyun Cho, and Jason Weston. 2019. Neu-
731 ral text generation with unlikelihood training. *arXiv*
732 *preprint arXiv:1908.04319*.

733 Daan Wierstra, Tom Schaul, Tobias Glasmachers,
734 Yi Sun, Jan Peters, and Jürgen Schmidhuber. 2014.
735 Natural evolution strategies. *The Journal of Machine*
736 *Learning Research*, 15(1):949–980.

737 Ronald J Williams. 1992. Simple statistical gradient-
738 following algorithms for connectionist reinforcement
739 learning. *Machine learning*, 8:229–256.

740 Kaichun Yao, Libo Zhang, Tiejian Luo, and Yanjun Wu.
741 2018. Deep reinforcement learning for extractive
742 document summarization. *Neurocomputing*, 284:52–
743 62.

744 Hongyi Yuan, Zheng Yuan, Chuanqi Tan, Wei Wang,
745 Songfang Huang, and Fei Huang. 2023. Rrhf: Rank
746 responses to align language models with human feed-
747 back. *Advances in Neural Information Processing*
748 *Systems*, 36.

749 Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho,
750 Sainbayar Sukhbaatar, Jing Xu, and Jason Weston.
751 2024. Self-rewarding language models. *arXiv*
752 *preprint arXiv:2401.10020*.

753 Susan Zhang, Stephen Roller, Naman Goyal, Mikel
754 Artetxe, Moya Chen, Shuohui Chen, Christopher De-
755 wan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022.
756 Opt: Open pre-trained transformer language models.
757 *arXiv preprint arXiv:2205.01068*.

758 Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Wein-
759 berger, and Yoav Artzi. 2019a. Bertscore: Eval-
760 uating text generation with bert. *arXiv preprint*
761 *arXiv:1904.09675*.

762 Wen Zhang, Yang Feng, Fandong Meng, Di You, and
763 Qun Liu. 2019b. Bridging the gap between training
764 and inference for neural machine translation. *arXiv*
765 *preprint arXiv:1906.02448*.

766 Yao Zhao, Mikhail Khalman, Rishabh Joshi, Shashi
767 Narayan, Mohammad Saleh, and Peter J Liu. 2022.
768 Calibrating sequence likelihood improves conditional
769 language generation. In *The Eleventh International*
770 *Conference on Learning Representations*.

771 Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B
772 Brown, Alec Radford, Dario Amodei, Paul Chris-
773 tiano, and Geoffrey Irving. 2019. Fine-tuning lan-
774 guage models from human preferences. *arXiv*
775 *preprint arXiv:1909.08593*.