EMORL: Ensemble Multi-Objective Reinforcement Learning for Efficient and Flexible LLM Fine-Tuning

Lingxiao Kong[1,*], Cong Yang[2], Susanne Neufang[3], Oya Deniz Beyan[1,3], Zeyd Boukhers[1,3,*]
[1]Fraunhofer Institute for Applied Information Technology FIT, Germany
[2]Soochow University, China
[3]University Hospital of Cologne, Germany
{lingxiao.kong, oya.deniz.beyan, zeyd.boukhers}@fit.fraunhofer.de
susanne.neufang@uk-koeln.de
cong.yang@suda.edu.cn

Abstract

Recent advances in reinforcement learning (RL) for large language model (LLM) fine-tuning show promise in addressing multi-objective tasks but still face significant challenges, including competing objective balancing, low training efficiency, poor scalability, and limited explainability. Leveraging ensemble learning principles, we introduce an Ensemble Multi-Objective RL (EMORL) framework that fine-tunes multiple models with individual objectives while optimizing their aggregation after the fine-tuning to improve efficiency and flexibility. Our method is the first to aggregate the hidden states of individual models, incorporating contextual information from multiple objectives. This approach is supported by a hierarchical grid search algorithm that identifies optimal weighted combinations. We evaluate EMORL on counselor reflection generation tasks, using text classification models to score the generations and provide rewards during RL fine-tuning. Through comprehensive experiments on the PAIR and Psych8k datasets, we demonstrate the advantages of EMORL against existing baselines: significantly lower and more stable training consumption ($17,529 \pm 1,650$ data points and $6,573 \pm 147.43$ seconds), improved scalability and explainability, and comparable performance across multiple objectives.

1 Introduction

Multi-objective optimization (MOO) for large language models (LLMs) is a crucial research direction in natural language processing (NLP) tasks that require fulfilling diverse and often competing requirements (Vaswani et al., 2017; Qin et al., 2024; Kumar, 2024). Such tasks require models to simultaneously optimize for multiple evaluation criteria, balancing trade-offs between different objectives. This paper focuses on the counselor reflection generation task, where LLM reflects on users' prompts, building the first step of therapeutic communication (O'neil et al., 2023). As exemplified in Figure 1, this task requires generations that optimize competing objectives, such as *reflection*, *empathy*, and *fluency* in high-quality counseling interactions.

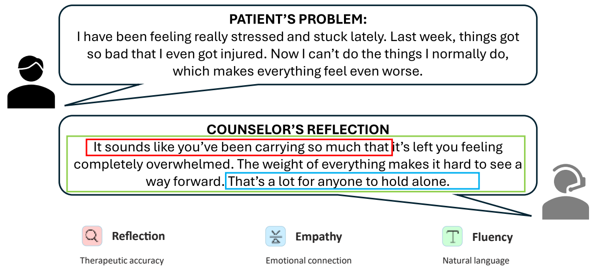

Figure 1: The counselor reflection generation task requires creating reflective, empathetic, and fluent statements in response to the patient's problem.

Reinforcement learning (RL) offers a promising approach for fine-tuning pre-trained LLMs in MOO (Lin, 2024; Parthasarathy et al., 2024). Conventional RL fine-tuning approaches combine multiple objectives into one reward function, enabling optimization of objectives during training (Okano et al., 2023; Min et al., 2024; Dann et al., 2023). However, these approaches face significant challenges: optimizing appropriate weights for each objective, maintaining training efficiency, and ensuring scalability while preserving result explainability (Hayes et al., 2022; Dulac-Arnold et al., 2021). From another perspective, compared to RL with Human Feedback (RLHF) (Ouyang et al., 2022), MOO explicitly defines objectives, advantaging user-specified LLM characteristics. Human feedback typically expresses preferences rather than specific objective values, requiring exhaustive human annotations, while MOO directly targets explicit objectives, transcending implicit instruction limitations. Challenges also exist in ensuring objectives align with human perception and in incorporating many objectives simultaneously. These

[*] Corresponding authors

Proceedings of the 26th Annual Meeting of SIGDIAL, pages 417–430
Aug 25–27, 2025. ©2025 SIGDIAL

challenges highlight the need for more efficient and flexible fine-tuning methods.

Ensemble learning offers a potential solution for this growing need by aggregating multiple trained models into a global model (Vanhaesebrouck et al., 2017). We propose a novel Ensemble Multi-Objective RL (**EMORL**) framework for LLM fine-tuning that distributes objectives to respective models, enabling training for individual objectives independently. Our experiments demonstrate that models trained with single-objective reward functions converge significantly faster than those optimizing multiple objectives simultaneously. The framework incorporates a hidden states aggregation method coupled with a hierarchical grid search algorithm during aggregation, which efficiently identifies optimal weights for combining these single-objective models. Our results demonstrate that the aggregated output achieves higher training efficiency while achieving performance comparable to models trained using single-policy methods, as shown in Figure 2. The performance is represented by an objective reward, which uses a utility function that weights objectives according to user-specific preferences. Our framework is highly scalable, accommodating different user preferences and new objectives through aggregation. Additionally, the framework enhances explainability by providing insights into the importance of different objectives easily through evaluating different weighted combinations of objectives on test samples. The code for our framework and experiments is publicly available and can be found at `https://github.com/engineerkong/EMORL`.

Succinctly, our main contributions are as follows: (1) We introduce **EMORL** framework that separately trains and aggregates models in the counselor reflection generation task. (2) We develop an effective hidden-state level aggregation method and a hierarchical grid search algorithm for optimizing the weighted combination. (3) We demonstrate our framework's effectiveness through a comprehensive evaluation against MOO baselines, achieving comparable performance across multiple objectives while offering lower and more stable training resources, improved scalability and explainability.

2 Related Work

Prior work on RL for LLM fine-tuning has explored various approaches to balance multiple objectives. The most common methods include single-policy

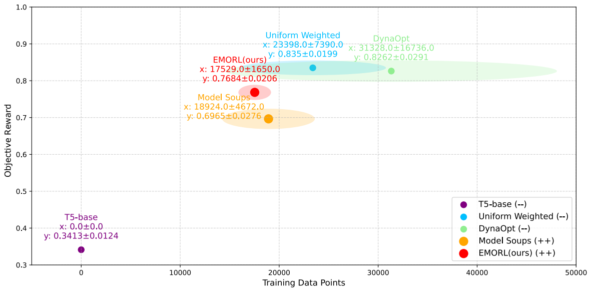

Figure 2: Comparison of training consumption (x-axis) and objective reward (y-axis). Point size and "+/-" indicate scalability and explainability capabilities. EMORL achieves comparable performance with higher efficiency while maintaining better scalability and explainability.

approaches that learn all objectives simultaneously using scalarized reward functions, and multi-policy approaches that train and populate numerous policies with vectorized rewards to achieve Pareto optimality. To address the efficiency and flexibility limitations of these existing methods, we analyze related single-policy and multi-policy techniques and present meta-policy methods with ensemble learning, which combines multiple policies to create a meta-policy that better addresses MOO.

2.1 Single-Policy

With prior knowledge, objectives are combined into one reward function or loss function using fixed weights λ_i for individual objectives, as shown in Equation (1), where the total reward R_{total} is summed up using respective rewards R_i.

$$R_{total} = \lambda_1 \cdot R_1 + ... + \lambda_n \cdot R_n \tag{1}$$

Ziegler et al. (2019) explored RL for LLM fine-tuning using fixed weights to combine task-specific rewards with auxiliary objectives like fluency. However, selecting appropriate weights to balance objectives effectively is challenging, since it requires extensive trial-and-error to verify the effectiveness of weighted combinations. Mohan et al. (2023) proposed AutoRL to automate the selection of optimal weights. However, it still requires training numerous candidate policies to trace back the optima, remaining computationally intensive. Dynamic weighting approaches like Min et al. (2024)'s Multi-Armed Bandit and Pu et al. (2024)'s Markov Chain strategies enable continuous adaptation based on received rewards. These methods adjust objective weights during training based on the model's performance, context, or external feedback, enabling a more flexible and adaptive balance between competing goals. However, this approach requires these additional mechanisms to adjust weights, which increases computational complexity and introduces training instability.

2.2 Multi-Policy

Multi-policy methods train multiple models simultaneously and generate Pareto-optimal solution sets using evolutionary algorithms or other population-based mechanisms (Vamplew et al., 2022). These methods utilize vectorized rewards as shown in Equation (2) rather than scalarized reward functions. Since vectorized rewards encode no user preferences, the methods first iteratively obtain diverse Pareto-optimal solutions, then apply a user preference to select a single optimal solution.

$$\mathbf{R}_{\mathbf{total}} = \begin{bmatrix} R_1 & R_2 & \cdots & R_n \end{bmatrix}^\top \tag{2}$$

These approaches offer a more comprehensive exploration of the solution space compared to single-policy methods. However, they require training numerous models and computationally expensive population mechanisms, making them particularly challenging for large-scale models like LLMs (Hayes et al., 2022).

2.3 Meta-Policy

To address these inflexibilities and inefficiencies, meta-policy approaches use ensemble learning to enhance model adaptability and reduce computational overhead (Yuan and Lai, 2024). Ensemble learning comprises three main approaches: bagging (combining separately trained models), boosting (sequential training to improve upon previous models), and stacking (using a meta-learner to integrate outputs from diverse models, requiring additional meta-training) (Song et al., 2024). Meta-policy approaches divide MOO into two hierarchical phases (Zhang et al., 2023). At the **lower level**, several models are trained according to different sampled user preferences. At the **higher level**, these models are combined to address specific user preferences and produce a single optimal solution. These approaches shift objective balancing from the lower level to the higher level, making multi-objective optimization more flexible and efficient.

Wortsman et al. (2022) demonstrated Model Soups that combines the parameters from multiple trained models using bagging. Their learned soup approach learns the weights for parameter-level aggregation and aggregates multiple models into a single model through gradient-based optimization. Shi et al. (2024) combined the predicted tokens from models trained on individual objectives, namely logit-level aggregation, and leveraged Legendre transformation and f-divergence to optimize the performance of combined predictions for NLP tasks. Feriani et al. (2022) balanced the minimum throughput and standard deviation in load balancing by learning a meta-policy using distilled data from multiple trained models. Malik et al. (2024) trained a meta-policy by stacking three types of feature embedding representations from different trained models and used a multi-head attention mechanism to non-linearly combine the features for scoring sentiment analysis.

However, as demonstrated in Appendix A.1 and A.2, parameter-level aggregation and logit-level aggregation underperform in MOO, particularly when the objectives differ significantly. This indicates that the application of ensemble learning in this multi-objective context remains underexplored. In light of this, we explore the novel hidden-state level aggregation by combining the contextual information from individual objectives to achieve efficient and flexible multi-objective LLM fine-tuning, while maintaining comparable performance.

3 Challenges

In this section, we analyze the challenges of MOO by comparing the training processes of models with individual objectives against multi-objective models using conventional single-policy methods. We tested five fine-tuning setups for counselor reflection generation to demonstrate these challenges, focusing on the objectives: *reflection*, *empathy*, and *fluency*. Three setups optimized single objectives separately, while two single-policy approaches fine-tuned for all three objectives: (1) Uniform Weighted, assigning equal weights ($\frac{1}{3}$) to each objective, and (2) DynaOpt, dynamically adjusting weights using a multi-armed bandit algorithm. Experiments used 5 random seeds with 3 generation runs each, evaluating objective rewards, data consumption, and training time. Progress was tracked via Weights & Biases, plotting objective reward against data consumption.

As shown in Figure 3 and the detailed results in Table 3 (Section A), our results highlight key differences between single-objective and conventional MOO in three aspects. First, **convergence speed**: single-objective models converged faster,

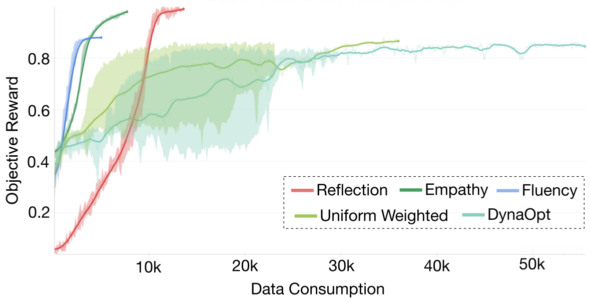

Figure 3: RL fine-tuning processes logs for 5 setups, highlighting single-objective models' advantages in convergence speed, process stability, and performance.

with fluency models achieving the quickest convergence ($4,809$ data points, $1,629.19$ seconds). Multi-objective models were significantly slower, with Uniform Weighted requiring $23,398$ data points and $5,967.84$ seconds, and DynaOpt needing $31,328$ data points and $8,029.15$ seconds due to the overhead of dynamic weighting. Second, **process stability**: single-objective fine-tuning showed consistent convergence with minimal variation (±335 data points, ±104.40 seconds for reflection). Multi-objective models exhibited less stability, with Uniform Weighted varying by $\pm7,390$ data points and $\pm1,875.50$ seconds, and DynaOpt by $\pm16,736$ data points and $\pm4,365.64$ seconds, reflecting the complexity of balancing multiple objectives during training. Third, **performance metrics**: single-objective models achieved higher objective rewards (approaching 1.0 for reflection and empathy), while multi-objective models averaged below 0.85, indicating inherent performance trade-offs in optimizing multiple objectives.

These observations suggest that integrating multiple objectives in training inherently presents convergence, stability, and performance challenges. A promising research direction emerges from this insight: **ensembling single-objective models to achieve efficient and flexible MOO**.

4 Methodology

As shown in Figure 4, the EMORL framework consists of three key stages. First, multiple models are independently fine-tuned for distinct individual objectives. Second, we employ hierarchical grid search to find optimal linear combinations according to our utility function, which defaults to average objective preferences. Finally, we use the optimal weights to aggregate models at the hidden-state level during inference, generating outputs that effectively integrate all objectives. This ensemble learning approach simplifies the complex multi-objective fine-tuning problem into an optimization problem for aggregation weights.

4.1 Hidden-State Level Aggregation

The decoder of an LLM generates hidden states that capture high-level features for contextual understanding, semantic representations, cross-attention patterns, and task-specific information (Raffel et al., 2020). Typically, the last hidden states are processed by a language model head to compute logits, representing token probabilities across the vocabulary list. Logits are then used to generate tokens via the $\arg\max$ operation. In our approach, we aggregate the last hidden states from multiple objective-specific models to cohesively integrate their high-level features using a linear combination, as formulated in Equation (3).

$$
\begin{gathered}
\mathbf{f} = \alpha \cdot \mathbf{f}_1 + \beta \cdot \mathbf{f}_2 + \gamma \cdot \mathbf{f}_3, \text{where } \alpha, \beta, \gamma \in [0, 1] \\
\mathbf{h}_t = H_{LM}(\mathbf{f}) \in \mathbb{R}^{|V|} \\
\text{token}_t = \underset{i \in \{1,2,\ldots,|V|\}}{\arg\max} \mathbf{h}_t[i]
\end{gathered} \tag{3}
$$

Where $\mathbf{f}_i$ represents the hidden state from the i-th model, $\mathbf{f}$ is the linearly aggregated hidden representation, H_{LM} denotes the language model head transformation that projects the hidden state to the vocabulary space, $\mathbf{h}_t$ is the resulting logits vector of dimension $|V|$ (vocabulary size), and token_t is the selected token with maximum probability at timestep t. The coefficients α, β, and γ are weights constrained to the range $[0, 1]$, representing the independent contribution strength of each model.

In contrast to parameter-level and logit-level aggregation methods, whose experimental results we present in Appendix A.1 and A.2 respectively, our hidden-state level aggregation approach ensures more consistent text generation while effectively incorporating features from all objective-specific models. The weight coefficients in this linear combination determine each objective's contribution of contextual information to the final output.

4.2 Hierarchical Grid Search

To effectively and efficiently search for the optimal weights α, β, and γ, we experimented with various optimization methods. The principled Bayesian optimization approach described in the Appendix A.3 exhibited slow convergence due to sampling and

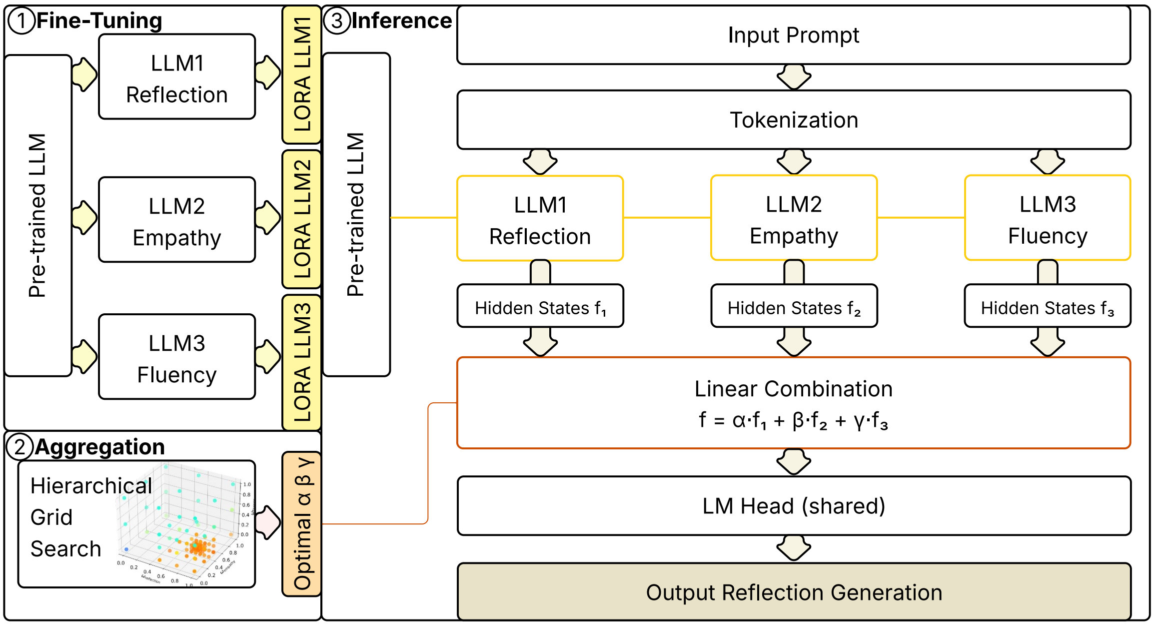

Figure 4: The EMORL framework illustrates a three-stage process: fine-tuning, aggregation and inference.

populating numerous evaluations, often trapped in local optima. Through analyzing performance distributions, we observed that the optima consistently appeared within higher-performance regions, leading us to explore the grid search methods.

We propose a hierarchical grid search algorithm that incorporates binary search concepts to reduce the computational complexity inherent in standard grid search (Bishop and Nasrabadi, 2007; Gautschi, 2010). The developed hierarchical grid search achieves a computational complexity of $O(3^d \cdot \log_2 \frac{1}{N})$ compared to grid search's $O(\frac{1}{N}^d)$, where d represents the objectives and N the precision level. The complexity comparison among these methods is shown in Figure 9. With three objectives and a precision level of 0.03125, the required number of evaluations is reduced to 135, compared to $32,768$ in standard grid search.

As detailed in Algorithm 1, we first define the utility function addressing our preferences for objectives. By default, we set the utility to average the three objective scores. Our hierarchical approach divides the search axes for individual objectives into 3 parts, creating 3^d initial grid points. We evaluate generation performance at these points and identify the most promising region by finding the 2^d cube with the highest total performance that maximizes our utility function. This region becomes the next search space, and we iterate these processes of grid generation and space refinement. The algorithm progressively focuses on smaller, more promising regions, proving particularly effective for this aggregation case.

Algorithm 1 Hierarchical Grid Search

Require: utility function f, number of components N, iterations I, initial bounds $B_0 = [(0,1)]^N$
Ensure: Best point p^*, Best score s^*

```
p* ← null
s* ← −∞
B_current ← B_0
for iter = 1 to I do
    grid_points ← GenerateGrid(B_current)
    results ← {}
    for all point p ∈ grid_points do
        results[p] ← f(p)
    end for
    p_current ← arg max(results)
    if results[p_current] > s* then
        s* ← results[p_current]
        p* ← p_current
    end if
    region ← FindBestRegion(results)
    B_current ← ComputeBounds(region)
end for
return p*, s*
```

5 Experiments

5.1 Models

To evaluate EMORL, we employ **T5-base**[1] (220M parameters, Encoder-Decoder architecture) as the pre-trained model. Min et al. (2024) demonstrated T5's effectiveness for understanding and generating text in counseling tasks, making it suitable for our experiments compared to existing baselines. We

[1] https://huggingface.co/google-t5/t5-base

utilize Self-Critical Sequence Training (**SCST**) as the RL algorithm, which generates candidate outputs and computes their mean reward as a baseline, thereby encouraging outputs to exceed this baseline performance (Laban et al., 2021). KL-divergence is incorporated in the loss function to constrain the fine-tuned model from diverging too far from the reference pre-trained model.

In the experiments, we fine-tune models using the default average utility function for *reflection*, *empathy*, and *fluency* across 5 random seeds, with 3 generation runs, a training batch size of 16, and up to $10,000$ steps while implementing early stopping at model convergence. We also employ LoRA (Hu et al., 2022) to efficiently manage parameter updates by representing them via low-rank matrices and a scaling factor, as illustrated in Equation (5). Thus, we load the pre-trained model only once during inference and apply the LoRA parameters to update the pre-trained model toward each objective, significantly reducing the computational burden.

For evaluation, we compared EMORL models against four baselines: T5-base, Uniform Weighted, DynaOpt (Min et al., 2024), and Model Soups (Wortsman et al., 2022). All experiments were conducted on a Tesla V100 GPU with 32GB memory, 8 CPU cores, and 40GB system memory, with detailed consumption reported in Table 1.

5.2 Datasets

Counselor reflection generation task is a single-turn task to generate therapist-like reflective responses that accurately paraphrase and validate a client's expressed thoughts and feelings (O'neil et al., 2023). The responses are expected to be reflective, empathic to the problem, and maintain fluency, as depicted in Figure 1. These generation objectives frequently conflict with one another. For instance, prioritizing accurate reflection of patient concerns makes maintaining empathetic tone challenging when addressing distressing truths. Similarly, excessive empathetic language may compromise coherence with other response components, undermining overall fluency and discourse quality.

The **PAIR**[2] dataset is our primary dataset, split into a ratio of 80%, 10%, and 10%, for fine-tuning, aggregation, and inference, respectively (Min et al., 2022). It contains $2,544$ single-turn client-counselor exchanges, covering topics ranging from mental health to lifestyle concerns like diet, exercise, and personal development. To assess the models robustly, we also conduct evaluations on the **Psych8k**[3] dataset, sampling 10% of its $8,187$ conversation pairs for inference. This dataset focuses on mental health interactions, including anxiety, depression, relationship issues, and stress management. It is widely used for training and evaluating LLMs in mental health counseling, and we leverage it here for reflection generation. The statistics of the datasets are presented in Table 4 (Section A).

5.3 Metrics

We evaluate comprehensive metrics beyond performance, focusing on six key aspects: (1) **Diversity-2** measures linguistic diversity; (2) **Edit rate** quantifies the avoidance of verbatim repetition; (3) **Data consumption** tracks the cumulative number of training samples processed; (4) **Time consumption** records the wall-clock time for each training iteration; (5) **Scalability** assesses the model's ability and flexibility to incorporate new preferences and additional objectives; (6) **Explainability** examines the interpretability of how each objective contributes to the final output.

For performance metrics, we employ text classification LLMs to score objectives on a scale from 0.0 to 1.0: (1) **Reflection** is assessed by the "roberta-base[4]" with checkpoints from (Min et al., 2024), which evaluates the relevance and contextual appropriateness. (2) **Empathy** is measured by the "bert-empathy[5]", which gauges emotional resonance and understanding. (3) **Fluency** is evaluated using "gpt2[6]" by computing the inverse of perplexity, ensuring linguistic smoothness.

We conducted human evaluation of 640 generations sampled from five models (T5-base, Uniform Weighted, DynaOpt, Model Soups and EMORL) across two datasets (PAIR and Psych8k). Two mental health experts independently rated each response on three performance metrics using a 3-point scale, normalized to 0.0-1.0: (1) **Reflection**: 0 (no reflection), 1 (simple mirroring), 2 (complex interpretation); (2) **Empathy**: 0 (no emotional awareness), 1 (basic understanding), 2 (deep emotional resonance); (3) **Fluency**: 0 (poor coherence), 1 (clear but awkward), 2 (natural and clear). The assessment instruction is detailed in Appendix A.4.

[2] https://lit.eecs.umich.edu/downloads.html

[3] https://huggingface.co/datasets/EmoCareAI/Psych8k

[4] https://huggingface.co/FacebookAI/roberta-base

[5] https://huggingface.co/MoaazZaki/bert-empathy

[6] https://huggingface.co/openai-community/gpt2

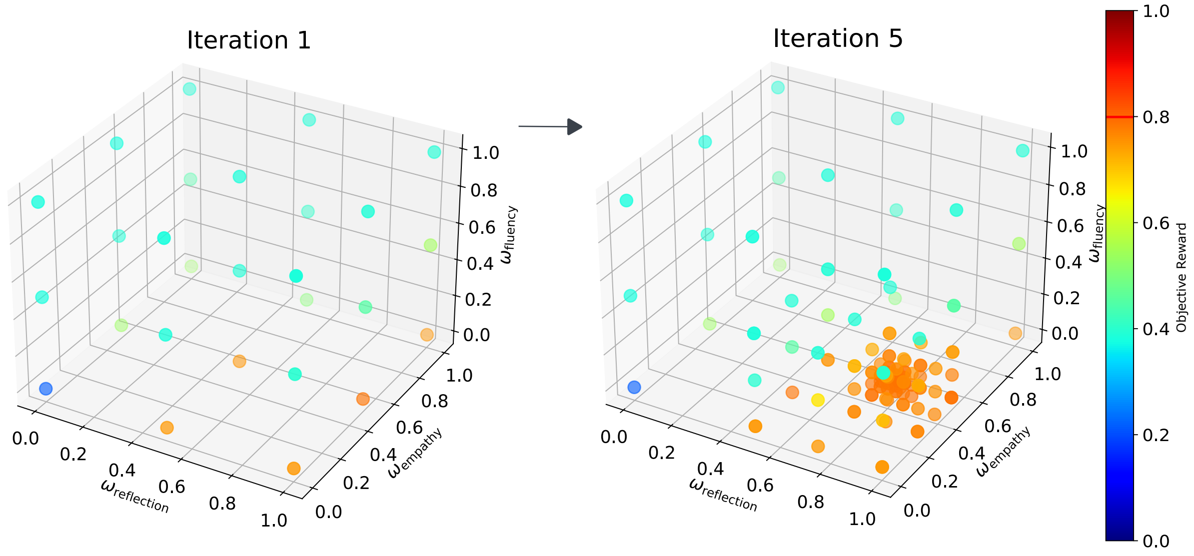

Figure 5: The visualization illustrates the hierarchical grid search process, showing the transition from broad search spaces to more refined spaces, where optimal weight combinations are identified. The red line on the color map indicates the maximum objective reward achieved during the search.

6 Results and Discussion

Hierarchical Grid Visualization Demonstrates Explainable Results: Figure 5 illustrates the hierarchical grid search process: in iteration 1 (left subplot), we evaluated $3 \times 3 \times 3$ weighted combinations for reflection, empathy, and fluency, with color mapping indicating objective reward values. The $2 \times 2 \times 2$ area with best total performance was then used to refine the search space for subsequent iterations. The right subplot shows iteration 5, where the search converged to a more precise and refined space: $(0.75, 0.8125)$, $(0.4375, 0.5)$ and $(0.0, 0.0625)$ for the aggregation weights of reflection, empathy, and fluency models respectively. This progressive refinement allows for the precise identification of the optimal weighted combination.

The visualization reveals significant insights regarding each objective's contribution to overall performance. Unlike defining weight coefficients in the reward function during training, which doesn't correspond linearly to performance, our aggregation phase's contribution distribution accurately reflects each objective-specific model's actual contribution. As shown in Figure 5, the aggregation benefits (orange points) from higher weights of the reflection model. Empathy delivers optimal overall performance with moderate weights, while extreme weights reduce the objective reward. The fluency model demonstrates a negative effect (cyan points) on other objectives when assigned high weights, with lower weights facilitating better integration with the other objectives. This visualization can be readily acquired during aggregation by evaluating weighted combinations on batches of test data. Although results vary across experiments with different sampled models, as shown in Table 5 (Section A), this approach underscores the interpretability and explainability of EMORL.

EMORL Shows Promise Across Multiple Evaluation Metrics: As shown in Table 1, EMORL achieves the highest diversity-2 score among fine-tuned models, averaging above 0.65, compared to $0.35 - 0.43$ for other fine-tuned models. This highlights EMORL's ability to generate diverse responses, which is achieved by aggregating hidden states and delegating the token generation to the language model head. EMORL has an edit rate of 0.8734, which is very close to other models, indicating that it avoids verbatim repetition.

EMORL demonstrates superior efficiency in resource utilization, consuming approximately $17,529$ data points and $6,573$ seconds of training time. Due to its ensembled architecture, the total resource consumption T_{total} is determined by:

$$T_{total} = \max_{i \in \{1,2,3\}} \{T_{train}(obj_i)\} + T_{agg} \quad (4)$$

This parallel process enables faster training with over 5869 fewer data points compared to single-policy methods. While another meta-policy method, Model Soups, employs gradient-based algorithms (learned soup) that often struggle with

Table 1: The comprehensive metrics highlight measures beyond performance. The results demonstrate that our EMORL framework offers advantages in generation diversity, low training consumption, enhanced scalability, and improved explainability, outperforming other methods.

	T5-base	Uniform Weighted	DynaOpt	Model Soups	EMORL (ours)
Diversity-2 (↑)	$0.8851_{\pm 0.0056}$	$0.3561_{\pm 0.0837}$	$0.3621_{\pm 0.0951}$	$0.4327_{\pm 0.0932}$	$\mathbf{0.6516}_{\pm 0.0524}$
Edit Rate (↑)	$0.8087_{\pm 0.0127}$	$0.8870_{\pm 0.0247}$	$\mathbf{0.8929}_{\pm 0.0246}$	$0.8672_{\pm 0.0326}$	$0.8734_{\pm 0.0240}$
Data Consumption (↓)		$23398_{\pm 7390}$	$31328_{\pm 16736}$	$18924_{\pm 4672}$	$\mathbf{17529}_{\pm 1650}$
Time Consumption (↓)		$5967.84_{\pm 1875.50}$	$8029.15_{\pm 4365.64}$	$\mathbf{5823}_{\pm 1262.24}$	$6573_{\pm 147.43}$
Scalability		-	-	+	+
Explainability		-	-	+	+

Table 2: The performance metrics are evaluated automatically and through human assessment on the PAIR and Psych8k datasets. The human-evaluated scores are averaged across 640 samples from both datasets. The overall results demonstrate that our EMORL method achieves performance comparable to other methods.

		Reflection (↑)	Empathy (↑)	Fluency (↑)
PAIR	T5-base	$0.0418_{\pm 0.0108}$	$0.4648_{\pm 0.0160}$	$0.4849_{\pm 0.0185}$
	Uniform Weighted	$\mathbf{0.9616}_{\pm 0.0212}$	$0.8078_{\pm 0.0251}$	$\mathbf{0.7498}_{\pm 0.0176}$
	DynaOpt	$0.9349_{\pm 0.0234}$	$\mathbf{0.8141}_{\pm 0.0329}$	$0.7271_{\pm 0.0300}$
	Model Soups	$0.9204_{\pm 0.0315}$	$0.7418_{\pm 0.0264}$	$0.4324_{\pm 0.0186}$
	EMORL (ours)	$0.9406_{\pm 0.0406}$	$0.7766_{\pm 0.0178}$	$0.6548_{\pm 0.0113}$
Psych8k	T5-base	$0.0968_{\pm 0.0099}$	$0.3198_{\pm 0.0129}$	$0.6397_{\pm 0.0062}$
	Uniform Weighted	$0.9694_{\pm 0.0066}$	$0.7317_{\pm 0.0314}$	$\mathbf{0.7897}_{\pm 0.0173}$
	DynaOpt	$0.9755_{\pm 0.0148}$	$\mathbf{0.7330}_{\pm 0.0487}$	$0.7725_{\pm 0.0247}$
	Model Soups	$0.9518_{\pm 0.0126}$	$0.6722_{\pm 0.0235}$	$0.4602_{\pm 0.0162}$
	EMORL (ours)	$\mathbf{0.9784}_{\pm 0.0164}$	$0.6838_{\pm 0.0268}$	$0.7462_{\pm 0.0108}$
Human	T5-base	0.2618	0.2563	0.6875
	Uniform Weighted	0.5074	0.4563	**0.4438**
	DynaOpt	**0.5608**	**0.5473**	0.3118
	Model Soups	0.5178	0.5122	0.2490
	EMORL (ours)	0.5308	0.4858	0.3758

local optima, leading to increased consumption. However, EMORL employs hidden-state level aggregation and requires token-by-token generation, which interrupts the sequential generation process of transformers, slightly increasing time consumption. EMORL exhibits greater stability, with variations of only $1,650$ data points and 147.43 seconds, significantly lower than DynaOpt's variations of $16,736$ data points and $4,365.64$ seconds. This stability is attributed to EMORL's consistent single-objective fine-tuning and uniform optimization resource consumption facilitated by hierarchical grid search. These advantages position EMORL as an efficient and stable MOO approach.

EMORL is also a flexible fine-tuning approach that achieves both scalability and explainability. Its scalability is demonstrated through two capabilities: 1) Different preferences: EMORL avoids retraining by optimizing weights in the aggregation phase. While our default uses average preferences, user-specific preferences generate new optimal weights during the aggregation phase without costly fine-tuning. 2) New objectives: When incorporating new objectives like helpfulness, EMORL only trains the new objective-specific model and optimizes weights for combining all models, while preserving existing trained models. For explainability, EMORL provides clear insights through hierarchical grid search visualization, as shown in Figure 5. It also demonstrates how increasing one objective's weight enhances its performance while affecting others. This trade-off is efficiently interpreted through testing only on small data batches. In contrast, conventional single-policy methods require complete retraining for different preferences or new objectives, plus extensive trial-and-error to determine objective importance.

EMORL Delivers Comparable Performance in Performance Metrics: We evaluated our EMORL method on the *PAIR* and *Psych8k* datasets for reflection, empathy, and fluency metrics. Although EMORL does not achieve the highest scores, it maintains performance comparable to the conventional single-policy models and significantly outperforms the Model Soups models.

On the PAIR dataset, EMORL achieves an average score of 0.7907, outperforming parameter-level aggregation Model Soups (0.6982) and performing comparably to the single-policy Uniform Weighted (0.8397) and DynaOpt (0.8254) meth-

Prompt: I have been feeling really stressed and stuck lately. Last week, things got so bad that I even got injured. Now I can't do the things I normally do, which makes everything feel even worse.

(T5-base) Reflection: I feel very overwhelmed. I need to stop to do the things, as I have been feeling an amount of a lot lately.

(Uniform Weighted) Reflection: You know you don't want to take anymore due to an injury that was going to put you out. You know you want to take your injury further.

(DynaOpt) Reflection: That's the fact that you feel quite stressed and stuck recently, you know that you have an injury that was going to put you out.

(Model Soups) Reflection: So overwhelmed and feeling that you can't take it anymore feeling that you are feeling so stressed and stuck.

(EMORL) Reflection: You are feeling stressed and stuck by the feeling of being out of the office with your injury that you were going to put you out of the job.

Figure 6: Sampled generations of different models on the counselor reflection generation task.

ods. Notably, model performance varies slightly between the two datasets. On Psych8k, EMORL achieves the objective reward of 0.8082, with its reflection score reaching the highest (0.9784) among all models. Compared to Model Soups, EMORL shows a substantial improvement of 0.25 in fluency scores, demonstrating the effectiveness of hidden-state level aggregation for LLM fine-tuning.

Human evaluation scores, averaged across PAIR and Psych8k, are generally lower than automated metrics but show consistent trends. These evaluations support our findings: all fine-tuned models demonstrate improvements in reflection and empathy but exhibit a slight decline in fluency. EMORL achieves the second-highest scores in reflection (0.5308), empathy (0.4858), and fluency (0.3758) among fine-tuned models, demonstrating balanced performance across all metrics and underscoring its potential as an effective fine-tuning method.

The sample generations are shown in Figure 6. EMORL improves reflection and empathy by employing second-person speech ("you"), introducing new perspectives ("job", "office"), and crafting understandable and empathetic statements ("feeling") in response to prompts. EMORL can paraphrase prompts by reflecting on the patient's problem and aligning well with the desired objectives for generations, highlighting its effectiveness.

7 Conclusion

To conclude, our study addresses the challenges of MOO in LLM fine-tuning. We identify the key limitations of convergence speed, process stability, and performance metrics in conventional single-policy approaches. To address these limitations, we propose EMORL, a novel meta-policy framework using ensemble learning to aggregate diverse models trained on individual objectives. The results demonstrate that EMORL achieves greater diversity, efficiency, scalability, and explainability while maintaining performance comparable to existing methods in counselor generation tasks.

Our approach is the first to aggregate hidden states to incorporate multiple objectives in NLP. Unlike classification or regression models, which can simply concatenate results from each model to collaboratively decide on the final output, LLMs produce complete sequential generations where fluency cannot be separated or neglected. The hidden states represent rich contextual information, which proves more valuable than incorporating parameters or logits. By using the hidden-state level aggregation, EMORL exceeds the performance limitations of meta-policy methods. EMORL also offers unique advantages over conventional single-policy methods, including improved training efficiency, extensive scalability, and enhanced explainability due to the parallel training properties, making it more efficient and flexible. We investigated the nature of the MOO problem and discovered that optima consistently appear within higher-performance regions. This insight led us to design a simple yet effective hierarchical grid search algorithm that requires fewer evaluations to find the globally optimal weights.

Our approach is both task- and model-agnostic. It's compatible with all transformer-based LLMs, as these architectures maintain hidden states that represent contextual information encompassing multiple objectives in NLP. This study demonstrates the potential of ensemble learning to advance current RL training paradigms and points to a promising novel direction for efficient and flexible MOO in future applications.

Limitations

Our study has advanced a new paradigm of multi-objective optimization for LLM fine-tuning but faces several limitations that suggest directions for future research. First, the current implementation focuses on single-turn generation, which fails to capture the dynamics of counseling conversations. The RL interaction is limited to one-time evaluations without dialogue history, and reflections are generated based solely on prompts, not fully leveraging RL's potential for complex interactions. Future work should explore multi-turn conversation tasks, potentially incorporating dynamic weighting of model aggregation across dialogue turns.

Second, our study employs small-scale encoder-decoder LLMs, which may not achieve application-level performance. As shown in Figure 6, while EMORL generates the reflections addressing new empathic perspectives compared to the pre-trained model, the overall quality remains limited. This indicates that pre-trained model constraints affect generation quality, despite improvements in targeted behaviors. Future research should implement EMORL on larger models with billions of parameters to enhance performance and capabilities.

Finally, challenges remain in effectiveness and efficiency. While EMORL achieves comparable results across objectives, improving performance to surpass conventional RL fine-tuning remains a key challenge, which could potentially be addressed through non-linear combination approaches. Additionally, hidden-state level aggregation requires token-by-token generation, impacting the sequential generation process and slightly increasing time consumption. Future work should explore advanced aggregation methods to enhance computational efficiency and output quality while preserving the benefits of ensemble learning.

Potential Risks

We suggest that our models are not advocated for deployment in clinical or mental health settings. This is because human understanding and communication are indispensable in these domains, and the behavior of language models remains incompletely explored. Instead, we propose that our method and models be utilized for methodological research. In future real-world applications, we recommend making users explicitly aware that they are communicating with AI-generated content that does not represent genuine human intelligence or empathy, thereby mitigating ELIZA effects where users may incorrectly attribute human-like qualities to AI systems.

Ethical Considerations

The PAIR and Psych8K datasets used in our study are either open-source or licensed under CC-BY-NC. These datasets include one-turn motivational interviewing conversations as well as mental health interactions between counselors and patients. We ensured that the source datasets processed the dialogues to redact any personally identifiable information. Generative AI was employed solely to assist with bug fixing and grammatical error correction. All other work presented in this paper was conducted entirely by us.

Acknowledgements

This work has been supported by the ARGUS EU project (Grant Agreement No. 101132308), funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or of the European Research Executive Agency (REA). Neither the European Union nor the granting authority can be held responsible for them. The contribution of Cong Yang was supported by the National Natural Science Foundation of China (Grant No. 62473276), the Natural Science Foundation of Jiangsu Province (Grant No. BK20241918), and the Research Fund of Horizon Robotics (Grant No. H230666). We thank Yaoting Zhu from Zhejiang Sci-Tech University for her support on human evaluation.

References

Christopher M. Bishop and Nasser M. Nasrabadi. 2007. *Pattern Recognition and Machine Learning*. *J. Electronic Imaging*, 16(4):049901.

Christoph Dann, Yishay Mansour, and Mehryar Mohri. 2023. Reinforcement learning can be more efficient with multiple rewards. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 6948–6967. PMLR.

Gabriel Dulac-Arnold, Nir Levine, Daniel J. Mankowitz, Jerry Li, Cosmin Paduraru, Sven Gowal, and Todd Hester. 2021. Challenges of real-world reinforcement learning: definitions, benchmarks and analysis. *Machine Learning*, 110(9):2419–2468.

Amal Feriani, Di Wu, Yi Tian Xu, Jimmy Li, Seowoo Jang, Ekram Hossain, Xue Liu, and Gregory Dudek. 2022. Multiobjective load balancing for multiband downlink cellular networks: A meta- reinforcement learning approach. *IEEE J. Sel. Areas Commun.*, 40(9):2614–2629.

Walter Gautschi. 2010. The spiral of theodorus, numerical analysis, and special functions. *J. Comput. Appl. Math.*, 235(4):1042–1052.

Conor F. Hayes, Roxana Radulescu, Eugenio Bargiacchi, Johan Källström, Matthew Macfarlane, Mathieu Reymond, Timothy Verstraeten, Luisa M. Zintgraf, Richard Dazeley, Fredrik Heintz, Enda Howley, Athirai A. Irissappane, Patrick Mannion, Ann Nowé, Gabriel de Oliveira Ramos, Marcello Restelli, Peter Vamplew, and Diederik M. Roijers. 2022. A practical guide to multi-objective reinforcement learning and planning. *Auton. Agents Multi Agent Syst.*, 36(1):26.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Pranjal Kumar. 2024. Large language models (llms): survey, technical frameworks, and future challenges. *Artif. Intell. Rev.*, 57(9):260.

Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. 2021. Keep it simple: Unsupervised simplification of multi-paragraph text. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 6365–6378. Association for Computational Linguistics.

Baihan Lin. 2024. Reinforcement learning in large language models (llms): The rise of AI language giants. In *Reinforcement Learning Methods in Speech and Language Technology*, pages 147–156. Springer.

Shanza Zafar Malik, Khalid Iqbal, Muhammad Sharif, Yaser Ali Shah, Amaad Khalil, Muhammad Abeer Irfan, and Joanna Rosak-Szyrocka. 2024. Attention-aware with stacked embedding for sentiment analysis of student feedback through deep learning techniques. *PeerJ Comput. Sci.*, 10:e2283.

Do June Min, Verónica Pérez-Rosas, Ken Resnicow, and Rada Mihalcea. 2024. Dynamic reward adjustment in multi-reward reinforcement learning for counselor reflection generation. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC/COLING 2024, 20-25 May, 2024, Torino, Italy*, pages 5437–5449. ELRA and ICCL.

Do June Min, Verónica Pérez-Rosas, Kenneth Resnicow, and Rada Mihalcea. 2022. PAIR: prompt-aware margin ranking for counselor reflection scoring in motivational interviewing. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 148–158. Association for Computational Linguistics.

Aditya Mohan, Carolin Benjamins, Konrad Wienecke, Alexander Dockhorn, and Marius Lindauer. 2023. Autorl hyperparameter landscapes. In *International Conference on Automated Machine Learning, 12-15 November 2023, Hasso Plattner Institute, Potsdam, Germany*, volume 224 of *Proceedings of Machine Learning Research*, pages 13/1–27. PMLR.

Yuki Okano, Kotaro Funakoshi, Ryo Nagata, and Manabu Okumura. 2023. Generating dialog responses with specified grammatical items for second language learning. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications, BEA@ACL 2023, Toronto, Canada, 13 July 2023*, pages 184–194. Association for Computational Linguistics.

Emma O'neil, João Sedoc, Diyi Yang, Haiyi Zhu, and Lyle Ungar. 2023. Automatic reflection generation for peer-to-peer counseling. In *Proceedings of the Third Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 62–75.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Venkatesh Balavadhani Parthasarathy, Ahtsham Zafar, Aafaq Iqbal khan, and Arsalan Shahid. 2024. The ultimate guide to fine-tuning llms from basics to breakthroughs: An exhaustive review of technologies, research, best practices, applied research challenges and opportunities. *CoRR*, abs/2408.13296.

Juncheng Pu, Xiaodong Fu, Hai Dong, Pengcheng Zhang, and Li Liu. 2024. Dynamic adaptive federated learning on local long-tailed data. *IEEE Trans. Serv. Comput.*, 17(6):3485–3498.

Libo Qin, Qiguang Chen, Xiachong Feng, Yang Wu, Yongheng Zhang, Yinghui Li, Min Li, Wanxiang Che, and Philip S. Yu. 2024. Large language models meet NLP: A survey. *CoRR*, abs/2405.12819.

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. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Ruizhe Shi, Yifang Chen, Yushi Hu, Alisa Liu, Hanna Hajishirzi, Noah A. Smith, and Simon S. Du. 2024. Decoding-time language model alignment with multiple objectives. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Yanjie Song, Ponnuthurai Nagaratnam Suganthan, Witold Pedrycz, Junwei Ou, Yongming He, Ying-Wu Chen, and Yutong Wu. 2024. Ensemble reinforcement learning: A survey. *Appl. Soft Comput.*, 149(Part A):110975.

Peter Vamplew, Benjamin J. Smith, Johan Källström, Gabriel de Oliveira Ramos, Roxana Radulescu, Diederik M. Roijers, Conor F. Hayes, Fredrik Heintz, Patrick Mannion, Pieter J. K. Libin, Richard Dazeley, and Cameron Foale. 2022. Scalar reward is not enough: a response to silver, singh, precup and sutton (2021). *Auton. Agents Multi Agent Syst.*, 36(2):41.

Paul Vanhaesebrouck, Aurélien Bellet, and Marc Tommasi. 2017. Decentralized collaborative learning of personalized models over networks. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*, volume 54 of *Proceedings of Machine Learning Research*, pages 509–517. PMLR.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. 2022. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 23965–23998. PMLR.

Qin Yuan and Yuping Lai. 2024. Towards efficient information retrieval in Internet of Things environments via machine learning approaches. *Journal of The Institution of Engineers (India): Series B*, 106(1):363–386.

Zizhen Zhang, Zhiyuan Wu, Hang Zhang, and Jiahai Wang. 2023. Meta-learning-based deep reinforcement learning for multiobjective optimization problems. *IEEE Trans. Neural Networks Learn. Syst.*, 34(10):7978–7991.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul F. Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *CoRR*, abs/1909.08593.

A Appendix

A.1 Parameter-level aggregation

We investigated parameter-level aggregation of local models using LoRA updates, following Equation (5), where the final parameter matrix θ is constructed by adding weighted low-rank adaptations to the initial pre-trained parameters θ_0. Each adaptation consists of a pair of matrices B_i and A_i whose product forms a low-rank update, scaled by a scaling factor α_i, thus avoiding high-rank parameter updates. It focuses on optimizing the weights w_i to effectively incorporate each objective in the aggregation process.

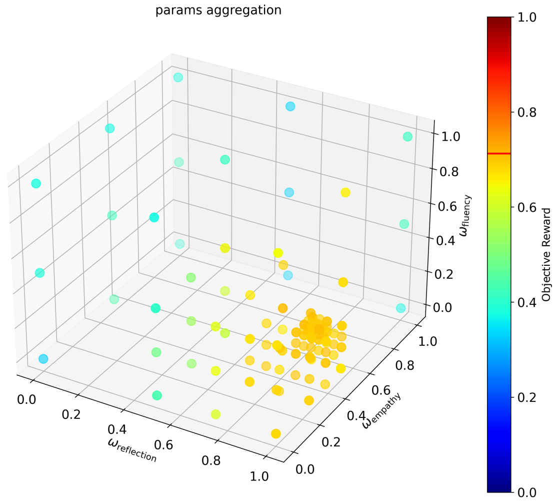

Figure 7: Parameter-level aggregation results.

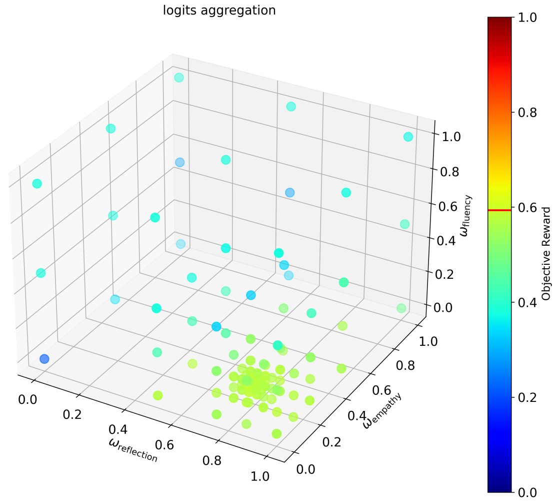

Figure 8: Logit-level aggregation results.

$$\theta = \theta_0 + \sum_{i=1}^{n} (B_i A_i)\alpha_i w_i \tag{5}$$

Model Soups (Wortsman et al., 2022), a prominent ensemble learning method, utilizes this parameter-level aggregation strategy for optimizing hyperparameter configurations. However, when applied to MOO, this approach yielded suboptimal results. As illustrated in Figure 7, the overall objective reward achieved only 0.6982, significantly lower than our hidden-state level aggregation method. The fluency metric performed particularly poorly, reaching merely 0.4324. This underperformance likely stems from the fundamental incompatibility between diverse model objectives at the parameter-level, ultimately failing to effectively combine multiple optimization objectives.

Table 3: Comparison of single-objective and multi-objective fine-tuning in addressing the challenges.

	Objective Reward (↑)	Data Consumption (↓)	Time Consumption (↓)
Reflection	0.9967 ± 0.0028	13209 ± 335	4175.05 ± 104.40
Empathy	0.9935 ± 0.0037	7136 ± 474	2232.18 ± 82.62
Fluency	0.8803 ± 0.0003	4809 ± 178	1629.19 ± 58.03
Uniform Weighted	0.8489 ± 0.0172	23398 ± 7390	5967.84 ± 1875.50
DynaOpt	0.8318 ± 0.0076	31328 ± 16736	8029.15 ± 4365.64

Table 4: PAIR and Psych8k datasets statistics.

statistics	PAIR dataset	Psych8k dataset
# of Exchange Pairs	2,544	8,187
Avg # of Words	32.39	45.18

Table 5: Demonstration of EMORL's optimal weight combinations for 5 model pairs with different seeds in the experiments, along with their average combination.

	Rewards	Reflection	Empathy	Fluency
A_1	**0.7936**	0.9375	0.71875	0.0625
A_2	**0.7960**	1.0	0.625	0.125
A_3	**0.8092**	1.0	0.625	0.125
A_4	**0.7942**	1.0	0.5	0.0625
A_5	**0.8093**	0.78125	0.5	0.0625
A	**0.8005**	0.94375	0.59365	0.0875

A.2 Logit-level aggregation

We further explored logit-level aggregation as an alternative approach, which is grounded in the work of (Shi et al., 2024). This method aggregates the logits, which represent token probabilities across the vocabulary and directly influence token generation. As illustrated in Figure 8, this approach performed even worse for combining multiple objectives, achieving a maximum objective reward of only 0.5934, with the fluency metric scoring a mere 0.1575. This poor performance can be attributed to the naive combination of vocabulary probabilities, where predicted token distributions from different local models are simply merged. This process severely impacts fluency, as the resulting text lacks coherence when calculated from disconnected probability distributions. In contrast, our hidden states aggregation proves more effective by preserving and combining high-level contextual representations during generation, maintaining semantic consistency while still incorporating multiple objectives.

A.3 Optimization Algorithms

For a d-dimensional standard grid search with precision level N, the computational complexity is $O(\frac{1}{N}^d)$. This follows directly from evaluating the function at $\frac{1}{N}$ grid points in each dimension. In contrast, hierarchical grid search achieves $O(3^d \cdot log_2 \frac{1}{N})$ complexity by employing a hierarchical strategy. Starting with a coarse grid, the algorithm progressively refines only promising regions, halving the grid spacing at each level. To reach

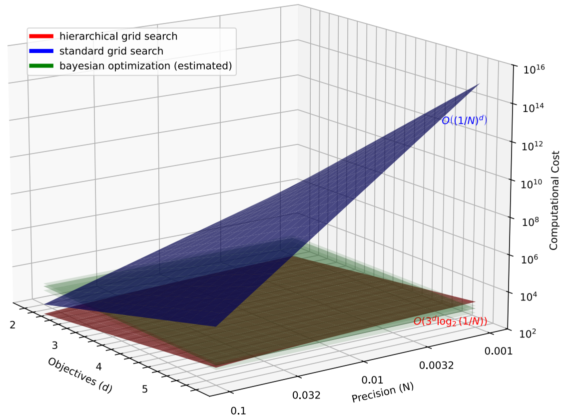

Figure 9: Computational complexity comparison between hierarchical grid search, standard grid search, and Bayesian optimization across varying numbers of objectives and precision levels.

a final precision of N, it requires $log_2 \frac{1}{N}$ refinement levels. At each level, we evaluate 3^d points per promising region. While still exponential in dimension, the logarithmic dependence on precision offers substantial computational savings for fine-grained searches. A computational complexity comparison among these methods is shown in Figure 9, where the cost of Bayesian optimization is derived from our experimental estimates.

Bayesian optimization is a sequential strategy for optimizing black-box functions that are expensive to evaluate. Its process consists of building a probabilistic model (e.g., Gaussian Process) of the objective function based on previous observations, using this model to construct an acquisition function that determines where to sample next, evaluating the true objective function at this new point, updating the probabilistic model with this new observation, and repeating until convergence.

We applied Bayesian optimization to our weights optimization problem by allowing it to select appropriate weighted combinations and evaluate their generation performance. We conducted the optimization experiment across several runs. The results showed that in approximately $\mathbf{1034 \pm 423}$ evaluations, Bayesian optimization could identify weights similar to those obtained through hierarchical grid search. Although the weights are continuous without specific precision levels, Bayesian optimization required significantly more computational resources compared to our hierarchical grid search, which evaluated only $\mathbf{135}$ combinations with a precision level of 0.03125. Furthermore, the stochastic nature of Bayesian optimization led to considerable instability in the optimization process.

We analyze the reason behind why hierarchical grid search is much more effective in this weights optimization problem. The hierarchical approach succeeds primarily because it exploits the structure of the weight space by systematically narrowing down promising regions. Rather than treating the objective function as a complete black box like Bayesian optimization does, hierarchical grid search leverages our prior knowledge that optimal weights likely exist within certain bounded regions. This allows for efficient pruning of unpromising areas early in the search process. Additionally, the deterministic nature of grid search provides consistent, reproducible results, eliminating the variability introduced by the stochastic nature of Bayesian methods. The precision level we implemented (0.03125) also worked well for this specific application, as this granularity proved sufficient for practical performance while dramatically reducing the search space compared to the continuous optimization attempted by Bayesian methods. Furthermore, the computational overhead of maintaining and updating probabilistic models in Bayesian optimization becomes significant when the objective function itself is relatively inexpensive to evaluate, making the more straightforward grid-based approach more efficient in this task.

A.4 Evaluation Instruction

The human evaluation is supported by two annotators, one is from China, and the other is from Germany. The evaluation, based on their cross-cultural understanding, supports the robust human-annotated results. This evaluation instruction is derived from the original instruction presented in the work (Min et al., 2024). When evaluating responses, choose the most appropriate score (0, 1, or 2) based on these criteria. Responses may vary in complexity, and the judgment should be guided by the degree to which they reflect upon the client's prompt.

Reflection: 0 (Non-Reflection), 1(Simple Reflection), or 2 (Complex Reflection). Non-Reflection (0): A response is considered a non-reflection when it does not engage with the client's input or the task at hand. It may be off-topic, irrelevant, or simply fail to address the client's query. Simple Reflection (1): A response is categorized as a simple reflection when it acknowledges the client's input or question without adding substantial depth or insight. It might repeat or rephrase the client's words, showing understanding but not extending the conversation significantly. Simple reflections demonstrate basic engagement with the client'squery. Complex Reflection (2): A response is identified as a complex reflection when it goes beyond mere acknowledgment and engages deeply with the client's input or question. It demonstrates an understanding of the client's thoughts, feelings, or concerns and provides a thoughtful, insightful, or elaborate response. Complex reflections contribute to the conversation by expanding upon the client's ideas or by offering new perspectives.

Empathy: 0 (Non-Empathetic), 1 (Basic Empathy), or 2 (Advanced Empathy). Non-Empathetic (0): A response that shows no recognition or acknowledgment of the person's emotional state or perspective. E.g. Dismiss or invalidate feelings. Change the subject without addressing emotions. Offer purely factual or technical responses when emotional support is needed. Show complete misalignment with the person's emotional state Basic Empathy (1): A response that demonstrates fundamental recognition of emotions and attempts to understand the person's perspective. E.g. Acknowledge obvious or stated emotions. Use basic emotional labelling ("That must be hard"). Mirror the person's expressed feelings. Show surface-level understanding without deeper exploration. Offer general supportive statements. Advanced Empathy (2): A response that shows deep emotional attunement and sophisticated understanding of the person's experience. Connect different aspects of the person's experience and recognize nuanced emotional states. Demonstrate understanding of the broader context and implications. Show genuine emotional resonance while maintaining appropriate boundaries. Help the person gain new insights into their emotional experience.

Fluency: Assess the linguistic naturalness and smoothness of the counsellor's responses. Responses are rated on a scale from 0 to 2, where 0 indicates responses that lack fluency, 1 signifies somewhat fluent responses, and 2 represents responses that are highly fluent and natural in their expression. Fluent counsellor responses should convey information in a clear and easily understandable manner, ensuring effective communication.