DMORL: Distributed Multi-Objective Reinforcement Learning Framework for Fine-Tuning Large Language Models in Counsellor Reflection Generation

Anonymous ACL submission

Abstract

Recent advances in reinforcement learning (RL) fine-tuning methods for large language models (LLMs) show promise in addressing multi-objective tasks but still have revealed significant challenges, including complex objective balancing, low training efficiency, poor scalability, and limited explainability. Leveraging distributed learning principles, we introduce a distributed multi-objective RL fine-tuning (**DMORL**) framework that simultaneously trains multiple models with individual objectives while optimizing their aggregation. Our method aggregates the last hidden states of local models to influence the final generation, supported by a hierarchical grid search algorithm that selects optimal weight combinations stepwise. This approach optimizes aggregation weights and significantly reduces the complexity of its selection process. We evaluate DMORL on a counsellor reflection generation task, using text-classification LLMs to score responses and reward RL fine-tuning. Through comprehensive experiments on the PAIR and Psych8k datasets, we demonstrate the advantages of DMORL 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

The multi-objective of large language models (LLMs) is a crucial research direction for natural language processing (NLP) tasks that need to fulfil diverse requirements (Vaswani, 2017; Qin et al., 2024; Kumar, 2024). This paper focuses on the counsellor reflection generation task, a crucial application of conversational AI in mental health, coaching, and counselling domains (O'neil et al., 2023). This task requires generating reflective responses that optimize multiple key objectives, such as reflection, empathy, and fluency in high-quality counselling interactions. We employ reinforcement learning (RL) to fine-tune pre-trained LLMs for these specific objectives (Lin, 2024; Parthasarathy et al., 2024). Conventional RL fine-tuning approaches combine multiple objectives into one reward function, enabling optimization of various and often conflicting objectives (Okano et al., 2023; Pérez-Rosas et al., 2024; Dann et al., 2023). However, these approaches face significant challenges: ensuring scalability as objectives increase, maintaining training efficiency, adapting models to dynamic conversations, and determining appropriate weights for each objective (Hayes et al., 2022; Dulac-Arnold et al., 2021). These challenges highlight the need for more flexible and efficient RL fine-tuning methods.

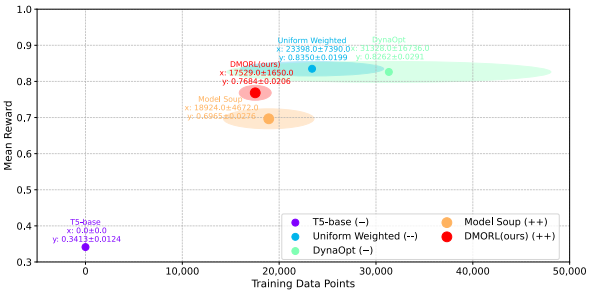

Figure 1: Comparison of training efficiency (x-axis) and mean reward (y-axis). Point size and "+/-" indicate scalability and explainability capabilities. DMORL achieves comparable mean rewards with lower data consumption while maintaining better scalability and explainability.

Distributed learning offers a potential solution for this growing need by training multiple local models and aggregating them into a global model (Vanhaesebrouck et al., 2017). We propose a novel distributed multi-objective RL fine-tuning (**DMORL**) framework for LLMs that operates in two phases: (1) We distribute objectives into multiple models to train individual objectives. The results demonstrate that the models with one ob-

jective in reward functions converge significantly faster than those with multiple objectives. (2) We introduce a states-level aggregation method and employ a hierarchical grid search algorithm during the aggregation to identify the optimal weights efficiently. Our experiments demonstrate that the aggregated output achieves performance comparable to models trained using conventional methods while achieving significantly higher training efficiency when considering the entire duration of the training and aggregation phases. This framework is also highly scalable, allowing additional objectives to be incorporated as modular components. Additionally, the framework enhances explainability by providing insights into the relative importance of different objectives. The code for our framework and experiments is publicly available (for research purposes only) and can be found at https://github.com/engineerkong/DMORL.

Succinctly, our main contributions are as follows: (1) We introduce a distributed multi-objective RL fine-tuning (**DMORL**) framework for counsellor reflection generation that separates training and aggregation phases. (2) We develop an efficient states-level aggregation method and a hierarchical grid search algorithm for weight optimization. (3) We demonstrate our framework's effectiveness through comprehensive evaluation against state-of-the-art baselines, such as 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.

2 Related Work

Prior work on RL fine-tuning of LLMs has explored various approaches to balance multiple objectives. These methods can be broadly categorized into several key directions, including conventional weighting methods and novel distributed approaches.

2.1 Fixed Weighting

This conventional method in multi-objective optimization involves explicitly defining and combining multiple objectives. In this approach, objectives are combined into one reward function or loss function using fixed weights, as shown in Eq 1, where λ_i are fixed weights. (Ziegler et al., 2019) explores RL fine-tuning for LLMs using fixed weights to combine task-specific rewards with auxiliary objectives like fluency and coherence. However, selecting appropriate weights to balance objectives effectively is challenging. (Mohan et al., 2023) proposed AutoRL to automate the selection of optimal weights. Nevertheless, AutoRL suffers from unexplainability and is computationally intensive as the number of objectives or hyperparameters grows.

$$R_{total} = \lambda_1 \cdot R_1 + ... + \lambda_n \cdot R_n \quad (1)$$

Fixed weighting methods are simple, interpretable, and easy to implement with existing optimization frameworks. However, they have limitations: (1) manual tuning of weights is required, which may not generalize across tasks or datasets; (2) static weights cannot adapt to dynamic conditions or contexts; and (3) extensive trial-and-error is needed to find optimal weights. To address these issues, researchers have explored dynamic approaches to weight adjustment in the reward function and the loss function. Alternatively, as proposed in this paper, the weight selection process can be shifted to the model aggregation phase, reducing computational overhead and complexity.

2.2 Dynamic Weighting

Dynamic weighting 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. Reward-driven approaches like (Pérez-Rosas et al., 2024)'s multi-armed bandit and (Pu et al., 2024)'s tabular MDP framework enable continuous adaptation based on received rewards. For gradient-based optimization, (Liu et al., 2020) employs GradNorm for weight adaptation, while (Ryu et al., 2024) uses gradient projection to resolve objective conflicts in summarization. Alternative formulations include (Zhou et al., 2024)'s RL-free approach using direct preference optimization, and (Jafari et al., 2024)'s Pareto surface optimization for prompt-based objectives.

Dynamic weighting is more flexible and adaptive and thus can handle changing conditions or tasks without manual intervention. Its weighting mechanism is eager to choose the objectives which have a higher probability of increasing the reward. However, this approach requires additional mechanisms to adjust weights, which increases computational complexity, and may introduce training instability. These limitations are evident in Figure 2, where the dynamic weighting method DynaOpt exhibits higher training costs and greater instability com-

pared to the fixed Uniform weighted.

2.3 Distributed Learning

Distributed learning enables training machine learning models across multiple devices or nodes (Li et al., 2020). This approach has proven effective in training models on diverse datasets, as demonstrated in federated learning (Zhang et al., 2021). While several studies have explored model aggregation at the parameters-level for multi-objective learning. (Wortsman et al., 2022) demonstrates Model Soup that combines fine-tuning, weight averaging, and validation in distributed learning to average weights from multiple fine-tuned models. Their learned soup approach leverages the concept of stacking to aggregate multiple models into a single model through gradient-based optimization. (Matena and Raffel, 2022) proposes merging fine-tuned models by weighting parameters based on Fisher information, which measures each parameter's importance for the task, effectively combining models trained for different objectives. (Zhu and Jin, 2019) demonstrates using multi-objective evolutionary algorithms to simultaneously optimize multiple competing objectives, balancing tradeoffs between objectives through Pareto optimization.

However, as demonstrated in Appendix 11.1, parameters-level aggregation underperforms in multi-objective optimization for NLP, particularly when the objectives differ significantly. It presents that the application of distributed learning in this context remains underexplored. In light of this, we analyze the challenges in the next section by comparing the training processes of single-objective models with multi-objective models using conventional weighting methods.

3 Challenges

We tested 5 fine-tuning setups for counsellor reflection generation to demonstrate the challenges, focusing on reflection, empathy, and fluency. Three setups optimized single objectives independently, while two multi-objective approaches combined all three objectives: (1) Uniform Weighted, assigning equal weights $(1/3)$ 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 mean rewards, data, and time consumption. Progress was tracked via Weights & Biases (W&B), plotting mean reward against data consumption.

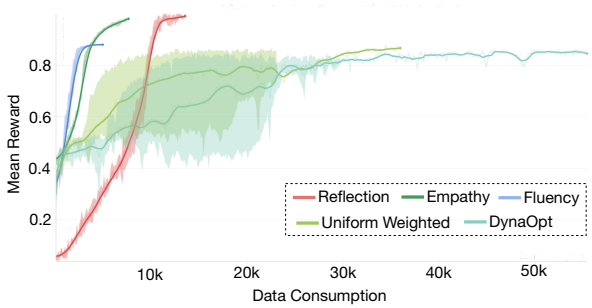

Figure 2: RL fine-tuning processes logs for 5 setups, highlighting single-objective models' advantages in convergence speed, process stability, and performance.

As shown in Figure 2 and Table 3, our results highlight key differences between single-objective and conventional multi-objective RL fine-tuning in three aspects. First, **convergence speed**: single-objective models converged faster, with fluency models achieving the quickest convergence ($4,809$ data points, $1,629.19$ seconds). Multi-objective models were 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 dynamic weighting. Second, **process stability**: single-objective fine-tuning showed consistent convergence with minimal variation (± 335 data points, ± 104.40 seconds for reflections). Multi-objective models were less stable, with Uniform Weighted varying by $\pm 7,390$ data points and $\pm 1,875.50$ seconds, and DynaOpt by $\pm 16,736$ data points and $\pm 4,365.64$ seconds, reflecting the complexity of balancing multiple objectives during the fine-tuning process. Third, **performance metrics**: single-objective models achieved higher 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.

This observation suggests a promising research direction: combining single-objective fine-tuned models through distributed learning to achieve multi-objective optimization. Beyond addressing challenges of convergence, stability, and performance, this approach could offer scalability by adding new objectives without retraining existing models and improved explainability by identifying which objective-specific models drive performance gains. Similar to multi-reward optimization, the goal is to find an effective compromise that leverages these potential benefits while delivering comparable performance.

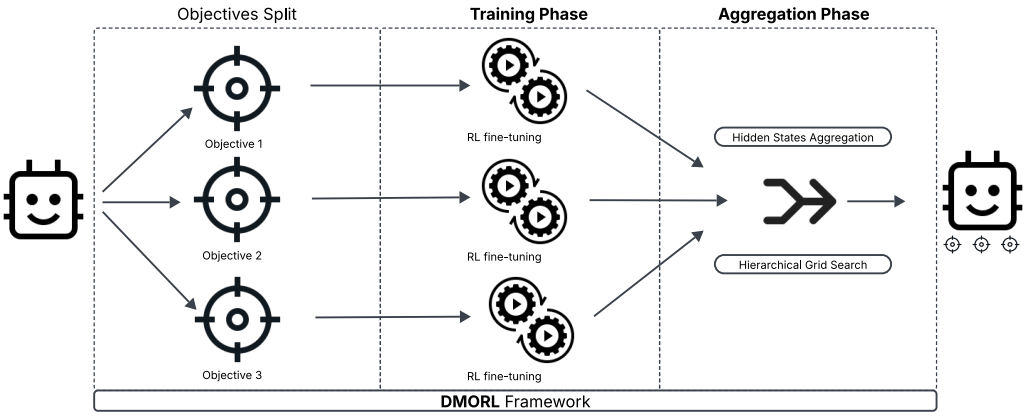

Figure 3: The DMORL framework illustrates the process of splitting objectives, training models with single objectives, and aggregating the models to achieve multiple objectives.

4 Methodology

Our DMORL framework splits objectives, trains single-objective models, and aggregates them using weight combinations, as shown in Figure 3. During training, multiple models are fine-tuned for individual objectives. In aggregation, we explore logits, parameters, and states-level aggregation, with logits and parameters-level aggregation produced suboptimal results as in Appendix 11, and states-level aggregation demonstrated promising performance. This distributed approach transforms multi-objective fine-tuning into optimizing aggregation weights, a well-studied mathematical problem. We address this optimization using an efficient hierarchical grid search algorithm that integrates grid and binary search principles (Bishop and Nasrabadi, 2006; Gautschi, 2011).

4.1 States-Level Aggregation

The decoder of LLMs generates hidden states capturing high-level features, including contextual understanding, semantic features, cross-attention patterns, and task-specific information (Raffel et al., 2020). Its last hidden states are processed by the language model head to compute vocabulary logits across $32,128$ tokens, which are then used to generate tokens via the $argmax$ operation, as illustrated in Eq 2. We aggregate the last hidden states to integrate high-level features cohesively. This approach ensures consistent text generation while incorporating features from all objective-specific models, with weights determining each model's contribution to the final output.

$$\begin{aligned} S_{agg} &= \sum_{i=1}^{n} w_i S_i \\ \text{logits} &= f_{lm\_head}(S_{agg}) \\ \text{token} &= argmax(\text{logits}) \end{aligned} \tag{2}$$

where $S_i \in \mathcal{R}^{b \times s \times h}$ is the last hidden state from model i, $w_i \in [0,1]$ is its weight, and b, s, h represent batch size, sequence length, and hidden dimension, respectively. f_{lm_head} projects the aggregated states to logits over the vocabulary.

4.2 Hierarchical Grid Search

As we transform the problem from multi-objective fine-tuning to aggregation weights optimization, the core challenge is to find the weights effectively and efficiently. The standard grid search method divides the search space into an N^d grid for d objectives, with N uniform divisions along each dimension, resulting in a computational complexity of $O(N^d)$. Gradient-based methods for determining weights are often inefficient, prone to converging to local minima, and require unstable trial-and-error processes to identify optimal solutions.

We combined grid search with binary search concepts and developed a hierarchical grid search algorithm, which improves the computational efficiency. With three objectives and a weight precision level of 0.03125, our approach reduced the number of evaluations to 135, compared to $32,768$ in standard grid search, yielding a computational complexity of $O(3^d \cdot \log_2 N)$. In this hierarchical approach, as shown in Algorithm 1, we first divide each search axis into 3 parts, creating 3^d initial grid points. We then evaluate the generation performance at these points and identify the most promising region by

finding the 2^d cube with the highest total performance score. This region becomes our next search space, and we iterate this process of grid generation and space refinement. The algorithm progressively focuses on smaller, more promising regions during iteration, which is effective for our aggregation case since the performance varies stably with respect to the weight combinations.

Algorithm 1 Hierarchical Grid Search

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

$p^* \leftarrow$ null
$s^* \leftarrow -\infty$
$B_{\text{current}} \leftarrow B_0$
for iter $= 1$ **to** I **do**
  grid_points $\leftarrow$ GenerateGrid(B_{current})
  results $\leftarrow \{\}$
  for all point $p \in$ grid_points **do**
    results$[p] \leftarrow f(p)$
  end for
  $p_{\text{current}} \leftarrow \arg\max$(results)
  if results$[p_{\text{current}}] > s^*$ **then**
    $s^* \leftarrow$ results$[p_{\text{current}}]$
    $p^* \leftarrow p_{\text{current}}$
  end if
  region $\leftarrow$ FindBestRegion(results)
  $B_{\text{current}} \leftarrow$ ComputeBounds(region)
end for
return p^*, s^*

5 Experiments

5.1 Model

We employ **T5-base**[1] (220M parameters) as it balances efficiency and performance for our distributed training approach. While larger models like GPT-3 or LLaMA might offer better raw performance, they would significantly increase computational costs in our multi-model setup. T5's encoder-decoder architecture also provides clear access to hidden states, critical for our states-level aggregation method. (Pérez-Rosas et al., 2024) demonstrated T5's effectiveness for counselling tasks, making it suitable for our experiments while enabling fair comparisons with existing baselines.

During the training phase, we optimizes reflection, empathy, and fluency for counsellor reflection generation. As the RL algorithm, we utilize Self-Critical Sequence Training (**SCST**), which generates candidate outputs and computes their mean reward as a baseline for encouraging candidates to outperform the mean reward (Laban et al., 2021). To ensure robustness, we train models using 5 random seeds per objective and conduct 3 runs per generation. Training is performed with a batch size of 16, validated every 8 steps using a validation batch size of 8. We employ LoRA (Hu et al., 2021) for efficient parameter updates, representing updates via low-rank matrices and a scaling factor.

During the aggregation phase, we pair three trained models with individual objectives, forming five model pairs. The LoRA parameters are loaded and applied to the T5-base model to construct the trained model. The validation dataset is then used to evaluate the performance of aggregated models with various weight combinations, with a batch size of 16. The best-performing weight combination is saved, allowing us to construct the aggregated model by combining the optimal weights and the LoRA parameters of the objectives.

For evaluation, we compare DMORL models against four baselines: T5-base, Uniform Weighted, DynaOpt and Model Soup models. The evaluation encompasses reward metrics for performance as well as evaluation metrics addressing additional aspects such as training efficiency. Additionally, two mental health experts assess the reward metrics for the generated responses. All experiments were conducted on a Tesla V100 GPU with 32GB memory, 8 CPU cores, and 40GB system memory.

5.2 Datasets

The **PAIR**[2] dataset is our primary dataset, split into training, validation, and testing sets with a ratio of $[80\%, 10\%, 10\%]$ (Min et al., 2022). It contains $2,544$ single-turn client-counsellor exchanges, covering topics ranging from mental health to lifestyle concerns like diet, exercise, and personal development. Each entry includes a client prompt and multiple reference responses.

To assess generalization, we also use the **Psych8k**[3] dataset, sampling 10% of its $8,187$ conversation pairs. This licensed dataset focuses on mental health interactions, including anxiety, depression, relationship issues, and stress management, and is widely used for training and evaluating LLMs in mental health counselling. Each

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

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

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

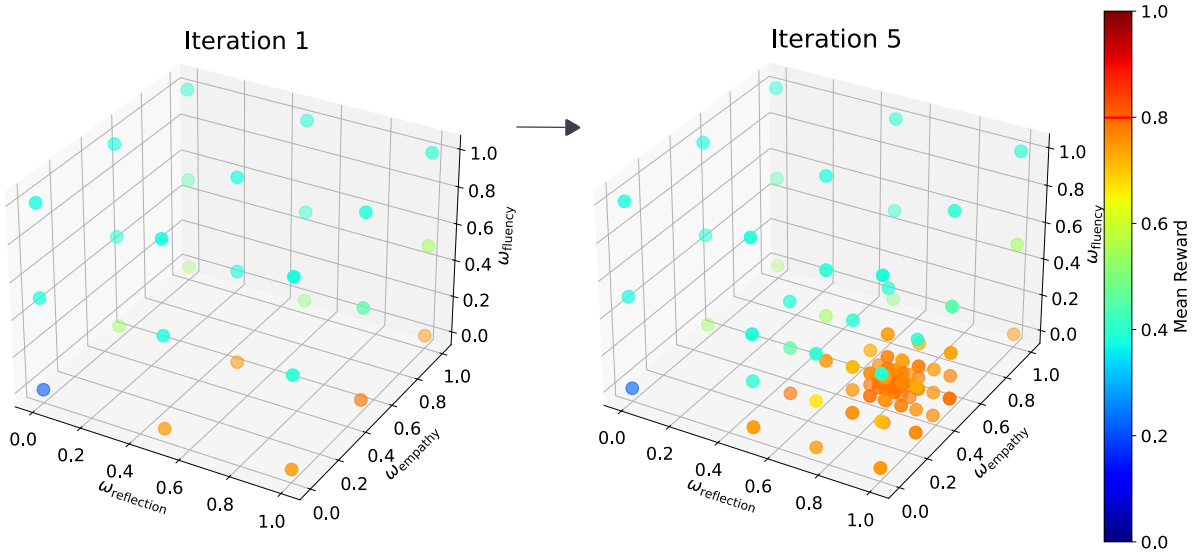

Figure 4: The visualization illustrates the hierarchical grid search process, showing the transition from a broad search space to a refined one, where optimal weight combinations are identified. The red line on the color map indicates the maximum mean reward achieved during the search.

entry consists of an instruction ("If you are a counsellor, please answer the questions based on the description of the patient."), a client input, and a counsellor's reference response.

5.3 Metrics

We evaluate multiple 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 to incorporate additional objectives; (6) **Explainability** examines the transparency of how each objective contributes to the final model.

For reward metrics, we employ specific LLMs to score three objectives on a scale from 0.0 to 1.0: (1) **Reflection** is assessed using the "roberta-base[4]", which evaluates the relevance and contextual appropriateness of responses. (2) **Empathy** is measured using 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 responses sampled from five models (T5-base, Uniform Weighted, DynaOpt, Model Soup and DMORL) across two datasets (PAIR and Psych8k). Two mental health experts independently rated each response on three reward 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).

6 Results Analysis

Hierarchical Grid Visualization Demonstrates Explainable Results. Figure 4 illustrates the hierarchical grid search process: in iteration 1 (left subplot), we evaluated $3 \times 3 \times 3$ weight combinations for reflection, empathy, and fluency, with color mapping indicating mean reward values. The top-performing $2 \times 2 \times 2$ combinations were then used to refine the search space for subsequent iterations. The right subplot of Figure 4 shows iteration 5, where the search converges 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 narrowing allows for the precise identification of optimal weight combinations.

The visualization also reveals important insights about each objective's contribution to the overall performance. The aggregation benefits from higher weights of the reflection model (orange points). Empathy delivers optimal overall performance with moderate weights, with both too-high and too-low

[4]https://huggingface.co/FacebookAI/roberta-base
[5]https://huggingface.co/MoaazZaki/bert-empathy
[6]https://huggingface.co/openai-community/gpt2

Table 1: The automated evaluation metrics on the PAIR dataset highlight additional measures beyond performance. They demonstrate that our DMORL framework offers advantages in generation diversity, low training consumption, enhanced scalability, and improved explainability, outperforming other methods.

	T5-base	Uniform Weighted	DynaOpt	Model Soup	DMORL (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		-	-	+	+

values reducing the mean reward. The fluency model demonstrates a negative effect on other objectives when assigned high weights (cyan points). Lower weights of the fluency model facilitate better integration with the other objectives (orange points in its low-weight regions). Although the results vary across aggregation experiments with different sampled models, the visualization underscores the interpretability and explainability of DMORL during the aggregation phase.

DMORL Shows Promise Across Multiple Evaluation Metrics. As shown in Table 1, DMORL 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 and 0.88 for the pre-trained model T5-base. This highlights DMORL's ability to generate diverse responses, which is achieved by aggregating hidden states and delegating the token generation to the model. While edit rates vary slightly among fine-tuned models, DMORL has the lowest rate but remains 0.07 higher than T5-base, indicating all models avoid verbatim repetition of client words.

DMORL demonstrates superior efficiency in resource utilization, consuming approximately 17,529 data points and 6,573 seconds of training time. Its distributed architecture, where total consumption is determined by $\max(c_{train}(obj_1), c_{train}(obj_2), c_{train}(obj_3)) + c_{agg}$, enables over 0.5× faster training compared to conventional multi-objective methods. Compared to another distributed learning method, Model Soup, DMORL maintains the lowest and most stable data consumption, as gradient-based algorithms in Model Soup often struggle with local optima, leading to increased consumption. However, DMORL's token-by-token generation using $model()$ slightly increases time consumption, making it higher than both the Model Soup and Uniform Weighted methods. DMORL also 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 DMORL's consistent single-objective training and uniform aggregation resource consumption facilitated by hierarchical grid search. These advantages position DMORL as an efficient and stable fine-tuning approach for multi-objective tasks.

DMORL achieves both scalability and explainability as another distributed learning method, Model Soup. Its scalability is evident when adding new objectives: instead of retraining the entire model, DMORL trains the new single-objective model and aggregates it with existing models. For explainability, DMORL provides clear insights through objective weight combinations and states-level aggregation patterns. As shown in Figure 4, the final hidden states reflect varying contributions: reflection at approximately 0.8, empathy at around 0.5, and fluency at about 0.05. Adjusting one objective's weight enhances its performance but may impact others, revealing trade-offs and quantifying each objective's contribution. In contrast, conventional multi-objective models lack scalability and interpretability, as they require extensive training to incorporate new objectives and rely on trial-and-error to determine the importance of each objective.

DMORL Delivers Comparable Performance in Reward Metrics. We evaluated our DMORL method on the PAIR and Psych8k datasets for reflection, empathy, and fluency. Although DMORL does not achieve the highest scores, it maintains performance comparable to the conventional multi-objective model and significantly outperforms both the T5-base model and the Model Soup model.

On the PAIR dataset, DMORL achieves a reflection score of 0.9106, only 0.05 below the best-performing Uniform Weighted model. Its empathy score is approximately 0.07 lower than DynaOpt, and its fluency score of 0.6248 is about 0.12 below the Uniform Weighted model. This indicates strong reflection capabilities but suggests room for improvement in fluency aggregation. On the Psych8k

Table 2: The reward metrics are evaluated automatically and through human assessment on the PAIR and Psych8k datasets. The human-evaluated scores are overall lower compared to the automated scores. In both evaluation approaches, the results demonstrate that our DMORL 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 Soup	$0.9204_{\pm 0.0315}$	$0.7418_{\pm 0.0264}$	$0.4324_{\pm 0.0186}$
	DMORL (ours)	$0.9106_{\pm 0.0406}$	$0.7466_{\pm 0.0178}$	$0.6248_{\pm 0.0113}$
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 Soup	$0.9518_{\pm 0.0126}$	$0.6722_{\pm 0.0235}$	$0.4602_{\pm 0.0162}$
	DMORL (ours)	$\mathbf{0.9784}_{\pm 0.0164}$	$0.6438_{\pm 0.0268}$	$0.7062_{\pm 0.0108}$
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 Soup	0.5178	0.5122	0.2490
	DMORL (ours)	0.5308	0.4858	0.3758

dataset, DMORL achieves the highest reflection score (0.9784), demonstrating strong generalization. However, its empathy and fluency scores remain 0.07-0.09 below the best models. This observation aligns with the aggregation weights shown in Figure 4: reflection at approximately 0.8, empathy at around 0.5, and fluency at about 0.05. The trade-off between objectives is evident, as increasing one weight often reduces performance in others. Model Soup, using parameters-level aggregation, performs poorly on both datasets, especially in fluency metrics (0.4463), indicating this approach is less effective and may hinder generation fluency.

Human evaluation scores are lower than automated metrics, indicating a discrepancy between LLM scoring objectives and real-world human perception. Nevertheless, human evaluation supports our findings: All fine-tuned models show improvements in reflection and empathy but a slight decline in fluency. DMORL achieves the second-highest scores in reflection (0.5308), empathy (0.4858) and fluency (0.3758) among fine-tuned models, demonstrating comparable performance across all metrics and underscoring its potential as an effective fine-tuning method. As shown in the sample generations in Figure 5, DMORL improves reflection and empathy by employing second-person speech and crafting responses that are understandable and empathetic to the prompts. However, the generated responses are not highly human-like, due to limitations of the T5 model and RL fine-tuning, where models tend to optimize toward extremes. Despite this, DMORL can generate responses that align well with the desired objectives, highlighting its competitive effectiveness.

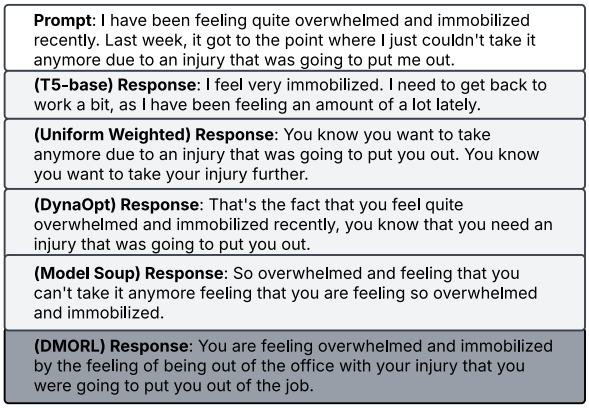

Figure 5: Sample reflection generations of different models on the counselor reflection generation task.

7 Conclusion

Our study addresses the challenges of conventional multi-objective RL fine-tuning in counsellor reflection generation, where multiple linguistic objectives are combined into one reward function. We identify challenges with convergence speed, process stability, and performance metrics and propose DMORL, a novel framework using distributed learning to train models on individual objectives. DMORL employs single-objective model training and states-level aggregation, transforming multi-objective fine-tuning into a weight optimization problem solved via hierarchical grid search. The results demonstrate that DMORL achieves greater diversity, efficiency, scalability, and explainability while maintaining performance comparable to existing baselines in counsellor generation tasks, bypassing the parameters-level aggregation methods. This demonstrates the potential of distributed learning to enhance RL fine-tuning processes.

8 Limitations

Our study has several limitations that suggest directions for future research. First, the current implementation focuses on single-turn generation, which does not capture the dynamics of real counselling conversations. The RL interaction is limited to one-time evaluations without dialogue history, and responses are generated based solely on prompts, failing to leverage RL's potential for complex interactions. Future work should explore multi-turn conversations, potentially incorporating dynamic weighting of model aggregation across turns.

Second, our study uses moderate-scale LLMs, which may not achieve practical application-level performance. As shown in Figure 5, while DMORL generates more responses addressing the second human's perspective compared to the pre-trained model, the overall quality remains limited. This indicates that baseline model constraints affect generation quality, despite improvements in targeted behaviors. Future research should implement DMORL on larger models with billions of parameters to enhance performance and capabilities.

Finally, challenges remain in effectiveness and efficiency. While DMORL achieves comparable results across objectives, improving performance to surpass conventional RL fine-tuning remains a key challenge. Additionally, states-level aggregation requires token-by-token generation, impacting processing speed. Future work should explore advanced aggregation methods to enhance computational efficiency and output quality while retaining the benefits of distributed learning.

9 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.

10 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 counsellors 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.

References

Christopher M Bishop and Nasser M Nasrabadi. 2006. *Pattern recognition and machine learning*, volume 4. Springer.

Christoph Dann, Yishay Mansour, and Mehryar Mohri. 2023. Reinforcement learning can be more efficient with multiple rewards. In *International Conference on Machine Learning*, 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.

Walter Gautschi. 2011. *Numerical analysis*. Springer Science & Business Media.

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

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

Yasaman Jafari, Dheeraj Mekala, Rose Yu, and Taylor Berg-Kirkpatrick. 2024. MORL-prompt: An empirical analysis of multi-objective reinforcement learning for discrete prompt optimization. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 9878–9889, Miami, Florida, USA. Association for Computational Linguistics.

Pranjal Kumar. 2024. Large language models (llms): survey, technical frameworks, and future challenges. *Artificial Intelligence Review*, 57(10):260.

Philippe Laban, Tobias Schnabel, Paul Bennett, and Marti A Hearst. 2021. Keep it simple: Unsupervised simplification of multi-paragraph text. *arXiv preprint arXiv:2107.03444*.

Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. 2020. Federated learning: Challenges, methods, and future directions. *IEEE signal processing magazine*, 37(3):50–60.

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.

Mingtong Liu, Erguang Yang, Deyi Xiong, Yujie Zhang, Yao Meng, Changjian Hu, Jinan Xu, and Yufeng Chen. 2020. A learning-exploring method to generate diverse paraphrases with multi-objective deep reinforcement learning. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2310–2321.

Michael S Matena and Colin A Raffel. 2022. Merging models with fisher-weighted averaging. *Advances in Neural Information Processing Systems*, 35:17703–17716.

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*, pages 148–158, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Aditya Mohan, Carolin Benjamins, Konrad Wienecke, Alexander Dockhorn, and Marius Lindauer. 2023. Autorl hyperparameter landscapes. *arXiv preprint arXiv:2304.02396*.

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 2023)*, pages 184–194.

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.

Venkatesh Balavadhani Parthasarathy, Ahtsham Zafar, Aafaq 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. *arXiv preprint arXiv:2408.13296*.

Verónica Pérez-Rosas, Ken Resnicow, Rada Mihalcea, et al. 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)*, pages 5437–5449.

Juncheng Pu, Xiaodong Fu, Hai Dong, Pengcheng Zhang, and Li Liu. 2024. Dynamic adaptive federated learning on local long-tailed data. *IEEE Transactions on Services Computing*.

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. *arXiv preprint arXiv: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. *Journal of machine learning research*, 21(140):1–67.

Sangwon Ryu, Heejin Do, Yunsu Kim, Gary Lee, and Jungseul Ok. 2024. Multi-dimensional optimization for text summarization via reinforcement learning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5858–5871, Bangkok, Thailand. Association for Computational Linguistics.

Paul Vanhaesebrouck, Aurélien Bellet, and Marc Tommasi. 2017. Decentralized collaborative learning of personalized models over networks. In *Artificial Intelligence and Statistics*, pages 509–517. PMLR.

A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. 2022. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International conference on machine learning*, pages 23965–23998. PMLR.

Chen Zhang, Yu Xie, Hang Bai, Bin Yu, Weihong Li, and Yuan Gao. 2021. A survey on federated learning. *Knowledge-Based Systems*, 216:106775.

Zhanhui Zhou, Jie Liu, Jing Shao, Xiangyu Yue, Chao Yang, Wanli Ouyang, and Yu Qiao. 2024. Beyond one-preference-fits-all alignment: Multi-objective direct preference optimization. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 10586–10613.

Hangyu Zhu and Yaochu Jin. 2019. Multi-objective evolutionary federated learning. *IEEE transactions on neural networks and learning systems*, 31(4):1310–1322.

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

11 Appendix

11.1 Parameters-level aggregation

We investigated parameters-level aggregation of local models using LoRA updates, following Eq 3, where A, B, and α represent the LoRA matrices and scaling factor respectively, and subscript i denotes the local model index. However, this approach yielded suboptimal results. As shown in Figure 6, the overall mean reward achieved only

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

	Mean 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: Demonstration of DMORL's best-performing weight combinations for 5 model pairs 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
$\bar{A}$	**0.8005**	0.94375	0.59365	0.0875

Table 5: PAIR and Psych8k datasets statistics.

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

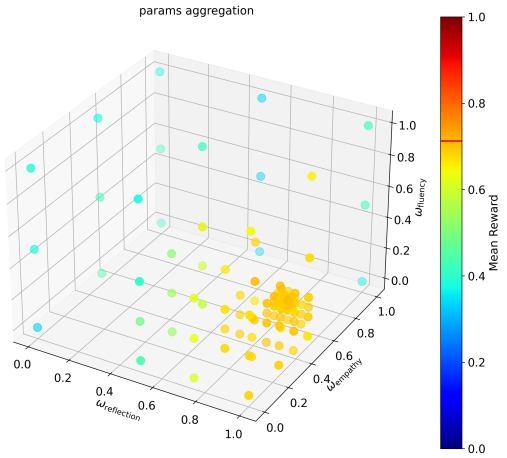

Figure 6: Parameters-Level Aggregation Results.

0.7123, notably lower than the states-level aggregation method. The fluency metric performed particularly poorly, reaching merely 0.4323. This underperformance likely stems from the fundamental difference in model objectives. Unlike federated learning, where local models share the same objective but train on different datasets, our local models are optimized for distinct objectives. Model Soup employs parameters-level aggregation by integrating various weight selection methods. However, this approach ultimately fails to effectively combine diverse objectives, highlighting the need for further research into more complex and effective parameters-level aggregation strategies.

$$\theta = \theta_0 + (B_1A_1)\alpha_1 w_1 + ... + (B_nA_n)\alpha_n w_n \quad (3)$$

11.2 Logits-level aggregation

We further explored logits-level aggregation as an alternative approach. Instead of combining the last hidden states, we aggregated the logits, which represent token probabilities across the 32,128-token vocabulary and directly influence token generation. As illustrated in Figure 7, this method performed even worse, achieving a maximum mean 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 tokens generated by different local models are simply concatenated. This process severely impacts fluency, as the resulting text comprises disconnected words calculated by different models. In contrast, the last hidden states aggregation proves more effective by preserving the high-level contextual information during generation.

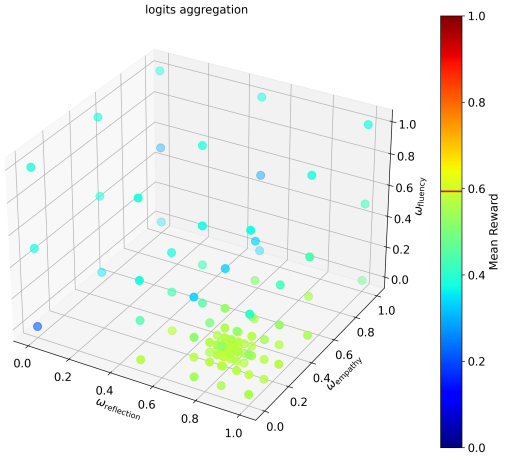

Figure 7: Logits-Level Aggregation Results.

11.3 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. 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.