Boosting LLM Translation Skills without General Ability Loss via Rationale Distillation

Anonymous ACL submission

Abstract

Large Language Models (LLMs) have achieved impressive results across numerous NLP tasks, and fine-tuning them for Machine Translation (MT) has improved their performance. However, vanilla fine-tuning often leads to catastrophic forgetting, compromising the broad general abilities of LLMs and introducing potential security risks. These abilities, which are developed using proprietary and unavailable training data, make simple data replay methods ineffective. To overcome this issue, we propose a novel approach called **RaDis** (**Ra**tionale **Dis**tillation). RaDis harnesses the strong generative capabilities of LLMs to create rationales for training data, which are then "replayed" to prevent forgetting. These rationales connect prior knowledge with new tasks, acting as self-distillation targets to regulate the training process. By jointly training on reference translations and self-generated rationales, the model can learn new translation skills while preserving its general abilities across other tasks. Additionally, RaDis provides a fresh perspective on using rationales in the CL field and has the potential to serve as a general continual learning method for a variety of tasks.

1 Introduction

Large Language Models (LLMs) have demonstrated exceptional performance across diverse Natural Language Processing (NLP) tasks but still fall short compared to conventional supervised encoder-decoder models in the realm of Machine Translation (MT). Recent studies have sought to enhance the translation performance of LLMs through continual instruction-tuning with parallel corpora (Yang et al., 2023; Xu et al., 2024). While this approach effectively boosts translation performance, it often introduces Catastrophic Forgetting (McCloskey and Cohen, 1989). As illustrated in Figure 1, fine-tuning instruction-tuned LLMs results in a significant decline in these models' performance on MT-Bench (Zheng et al., 2023), indicating a loss in general instruction following abilities. Such degradation not only limits the broad application of LLMs but also hinders their translation performance, such as formality steering and contextualization capabilities (Stap et al., 2024).

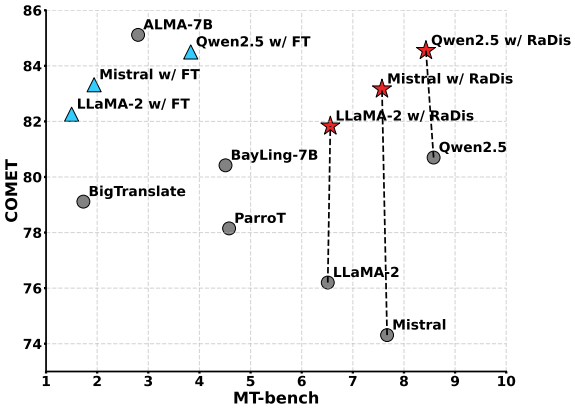

Figure 1: Translation performance (COMET) and general conversational and instruction-following ability (MT-Bench). While both Fine-tuning (blue triangle) and RaDis (red star) greatly enhance the translation performance, RaDis helps preserve most of the models' general ability.

While various replay-based approaches have been proposed to mitigate forgetting (Mok et al., 2023; He et al., 2024; Wang et al., 2024), these methods do not apply to our target scenario. This limitation arises because the superior performance of LLMs largely depends on high-quality, proprietary data, to which downstream users have no access for replay. Previous studies have explored the use of open-source monolingual data (Stap et al., 2024) or instruction-following datasets (Jiao et al., 2023; Zhang et al., 2023; Alves et al., 2024) as alternatives. However, their effectiveness remains limited as open-source datasets struggle to match the performance of proprietary ones.

To address this problem, this paper explores leveraging the strong generative ability of LLMs

to *synthesize their own replay data*. However, given the vast task space of LLMs and the limited data we could use, generating high-quality synthesis data that encapsulates diverse general knowledge remains a non-trivial question. We found that instruction-tuned LLMs are capable of generating detailed **rationales** when tasked with translation requests (see Section 3.1 and Appendix F for more details). Further analysis suggests that these rationales encapsulate the internal knowledge LLMs utilize during translation. Building upon these findings, we propose a novel training method named **RaDis** (**Ra**tionale **Dis**tillation). It prompts the LLM to generate rationales for the reference translations in MT training data and then concatenates references and rationales, forming an enriched dataset for subsequent training. By incorporating both the rationales and the references into the training, RaDis ties LLMs' internal knowledge with new tasks to learn and introduces a self-distillation loss on the rationale, thereby mitigating the forgetting issue.

Comprehensive experiments using three widely adopted LLMs, LLaMA-2-7B-Chat (Touvron et al., 2023), Mistral-7B-Instruct-v0.2 (Jiang et al., 2023), and Qwen2.5-7B-Instruct (Yang et al., 2024a) validates the effectiveness of RaDis. As depicted in Figure 1, RaDis improves translation performance to a level comparable to vanilla fine-tuning while maintaining the model's original proficiency on general ability benchmarks. Further analysis reveals that distilling self-generated rationales not only outperforms distilling from external rationales generated by a stronger model but also avoids the conflict between learning new tasks and consolidating the original ability. Together, these findings offer additional insights into RaDis' effectiveness and future study.

In summary, this work makes the following contributions:

- It proposes RaDis, a novel training method that enhances LLMs' translation proficiency while preserving their generality. RaDis uses the inherent ability of LLMs to generate rationales for translation data (Section 3.1) and performs self-knowledge distillation on these rationales to alleviate forgetting (Section 3.3).
- RaDis addresses a critical challenge in LLM application. In practice, LLMs are usually fine-tuned for downstream tasks. This process suffers from catastrophic forgetting, and the replay-based approach is not suitable due to the absence of the training data of LLMs. RaDis overcomes this problem by replaying self-generated rationales, which provides a fresh angle in the field of CL.
- Experimental results show that RaDis significantly outperforms baselines. Most notably, it achieves translation performance comparable to vanilla fine-tuning while preserving 92% of the LLM's original ability. Further analysis demonstrates that RaDis avoids optimization conflict and generalizes well to broader tasks.

2 Related Works

2.1 Fine-tuning LLMs for MT

LLMs have achieved promising results in MT (Zhu et al., 2024; Guo et al., 2024; Song et al., 2025). Previous studies have primarily fine-tuned LLMs using parallel corpora to enhance their translation proficiency (Yang et al., 2023; Xu et al., 2024; Zheng et al., 2024a; Yu et al., 2025; Lu et al., 2024). Although these methods enhance the translation proficiency of LLMs, they often compromise the models' general ability and internal knowledge (Stap et al., 2024). To address this issue, several studies have proposed to add open-source monolingual data (Stap et al., 2024) or instruction-following data (Jiao et al., 2023; Zhang et al., 2023; Alves et al., 2024) into fine-tuning. However, limited by the quality of the data and the distribution gap, they still suffer from forgetting and underperforming open-sourced LLMs by a large margin. In contrast to these efforts, our approach solely uses machine translation data and can preserve the general ability by distilling self-generated rationales.

2.2 Continual Instruction Tuning

Continual instruction tuning (CIT) seeks to mitigate CF during the instruction tuning of LLMs by employing CL approaches (Wu et al., 2024; Shi et al., 2024). Traditional CL methods are typically divided into replay-based, regularization-based, and architecture-based methods (Ke and Liu, 2022). However, in the context of LLMs, the vast parameter and task space reduces the feasibility of regularization-based and architecture-based methods (Wang et al., 2024). As a result, current research has predominantly focused on replay-based techniques and their variants (Scialom et al., 2022; Yin et al., 2022; Mok et al., 2023; He et al., 2024; Wang et al., 2024) While these approaches are

promising, they are subject to the reliance on access to the original training data. Consequently, they cannot be applied to mitigate the forgetting of instruction-tuned LLMs' general abilities gained from in-house training data. SDFT (Yang et al., 2024b) is the first work designed for preserving the general instruction-following abilities of LLMs. It proposes to paraphrase the original train dataset with the LLM itself to bridge the distribution gap. However, the quality of the paraphrased data is limited by the capabilities of the prompt and the model itself, which may diminish the performance of the task to be learned. In contrast, RaDis argues the original data with self-generated rationales and avoids loss of performance on new tasks.

3 Method

We begin by presenting a key observation: when tasked with translation requests, instruction-tuned LLMs can generate detailed **rationales** that encapsulate the internal general knowledge leveraged during translation (Section 3.1). Building on this insight, we introduce **Ra**tionale **Dis**tillation (RaDis), which leverages these self-generated rationales as replay data to help the model retain its broad general capabilities (Section 3.1). Finally, we demonstrate that the RaDis training objective can be decomposed into a conventional MT loss and a self-distillation loss on rationale tokens, which helps prevent excessive deviation of model parameters (Section 3.3).

3.1 Observation: Self-generated Rationales

Figure 2: An example of LLM's response to translation instruction that encompasses a rationale.

Instruction-tuned LLMs exhibit a strong ability to follow instructions and engage in conversational interactions, delivering helpful responses

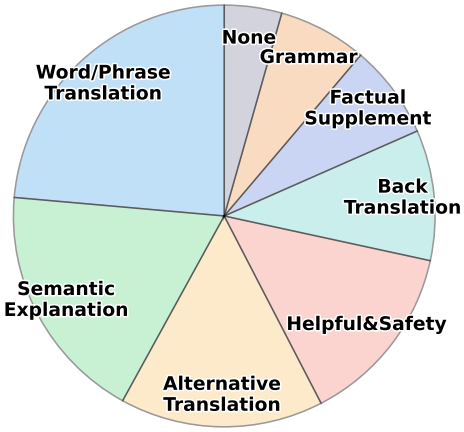

Figure 3: Distribution of translation rationales.

across a wide range of tasks. Unlike conventional models that simply output the answer, instruction-tuned LLMs are known to be able to generate rationales (Wei et al., 2022). As illustrated in Figure 2, when presented with a translation request, instruction-tuned LLMs not only generate the translation but also provide an accompanying rationale.

We randomly sampled 250 sentences from diverse translation directions and extracted their corresponding rationales for analysis. The content of these rationales included diverse information, as expected. As shown in Figure 3, rationales were generated for more than 95% of the sentences. These rationales can be broadly categorized into seven types, ranging from word alignments to factual knowledge.[1] Intuitively, these rationales preserve the original knowledge in LLMs. Building upon this insight, we propose our approach, named RaDis.

3.2 RaDis: distilling rationales to alleviate forgetting

The forgetting issue can be attributed to an unsuitable training approach. In conventional fine-tuning, the supervision signal comes from the reference sentence solely, which biases the model to a translation-specific distribution. Previous studies have sought to address this issue by replay-based methods. However, due to the absence of original training data for LLMs and the poor quality of open-sourced instruction-following data, their effectiveness is limited. To this end, we propose RaDis. The core idea of RaDis is similar to *pseudo-replay*, which employs an additional data generator to synthesize replay data (Shi et al., 2024). How-

[1]For examples of rationales, please refer to Appendix F.

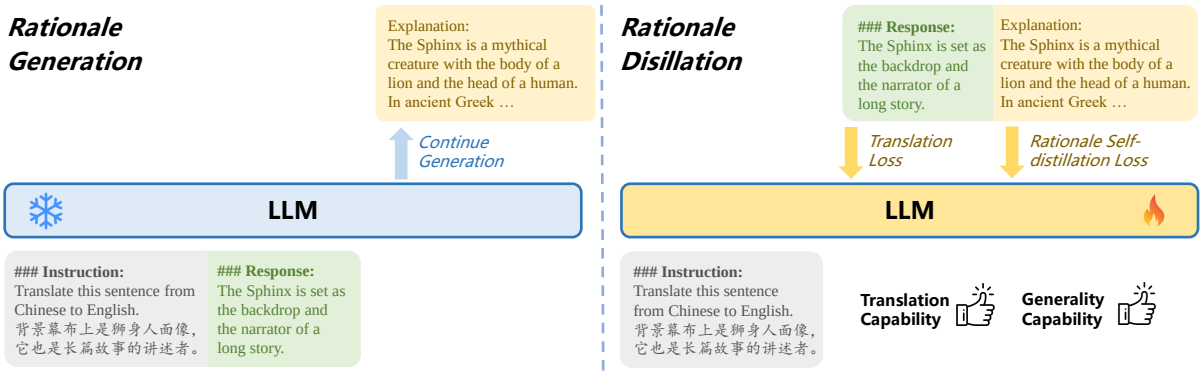

Figure 4: Overview of the RaDis approach. **Rationale Generating (Left)**: Given a translation instruction-response pair as an input, the LLM extends the response by generating a rationale. **Fine-tuning with Rationale Distillation (Right)**: RaDis utilizes this self-generated rationale to enrich the original response and fine-tunes the LLM with the enriched data. The CLM loss computed on the rationale serves as a self-distillation regularization term, preventing excessive parameter divergence.

ever, the superior generative abilities of LLMs now allow us to *leverage the model itself to synthesize this replay data*. As depicted in Figure 4, RaDis starts from an instruction-tuned LLM as the backbone. It utilizes a prompt template $\mathcal{I}$ to format the translation sentence pair (x, y) and sends them into the backbone LLM parameterized as θ. As shown in Section 3.1, LLMs have the inherent ability to continue generating a rationale $\mathbf{r}$ using the translation instruction-response pair as the prefix.

$$\mathbf{r} \sim P(\mathbf{y}, \mathbf{x}, \mathcal{I}) \tag{1}$$

These rationales encapsulate the internal general knowledge leveraged during translation, building a "semantic scaffold" that ties previous knowledge with new tasks. Therefore, introducing them into training improves knowledge retention. Specifically, the self-generated rationale $\mathbf{r}$ is concatenated with the translation sentence $\mathbf{y}$, creating an enriched response $\hat{\mathbf{y}} = \text{CONCAT}(\mathbf{y}, \mathbf{r})$. The enriched instruction-response pair is subsequently used to train the backbone LLM using a standard causal language model (CLM) loss, defined as:

$$\mathcal{L}(\mathbf{x}, \hat{\mathbf{y}}) = -\log P(\hat{\mathbf{y}}|\mathbf{x}, \mathcal{I}) \tag{2}$$

The enriched response now incorporates both the task-specific knowledge for translation and diverse original knowledge embedded within the self-generated rationale. As a result, fine-tuning the model with it can learn the translation task and consolidate the original general ability simultaneously.

3.3 Why RaDis Works: A Knowledge Distillation Perspective

In previous sections, we discovered that self-generated rationales are effective substitutes for replay data. However, since they are neither traditional replay data nor pseudo-replay data, a natural question arises: why do they work? Here, we demonstrate that RaDis can be understood as a form of knowledge distillation, a technique proven to mitigate forgetting. To explain this, we paraphrase Equation 2 as:

$$\begin{aligned}\mathcal{L}(\mathbf{x}, \hat{\mathbf{y}}) &= -\log P(\hat{\mathbf{y}}|\mathbf{x}, \mathcal{I}) \\ &= -\sum_{t=1}^{T+R} \log P(\hat{\mathbf{y}}_t|\hat{\mathbf{y}}_{<t}, \mathbf{x}, \mathcal{I})\end{aligned} \tag{3}$$

where R is the length of the rationale $\mathbf{r}$. The properties of the CLM loss allow us to split and reassemble the loss across each token. By separating the loss of the reference translation from the self-generated rationale, we obtain:

$$\begin{aligned}\mathcal{L}(\mathbf{x}, \hat{\mathbf{y}}) &= -\sum_{t=1}^{T+R} \log P(\hat{\mathbf{y}}_t|\hat{\mathbf{y}}_{<t}, \mathbf{x}, \mathcal{I}) \\ &= -\sum_{t=1}^{T} \log P(\mathbf{y}_t|\mathbf{y}_{<t}, \mathbf{x}, \mathcal{I}) \\ &\quad -\sum_{t=T+1}^{T+R} \log P(\mathbf{r}_t|\mathbf{r}_{<t}, \mathbf{y}, \mathbf{x}, \mathcal{I}) \\ &= -\log P(\mathbf{y}|\mathbf{x}, \mathcal{I}) - \log P(\mathbf{r}|\mathbf{y}, \mathbf{x}, \mathcal{I})\end{aligned} \tag{4}$$

Here, the first term is a traditional MT loss, which trains the model to acquire new translation knowledge. The second term minimizes the negative log-likelihood of the self-generated rationale $\mathbf{r}$ given the conventional translation instruction-response pair. It can be interpreted as a sequence-level self-distillation loss with the rationales as the distillation target. In Section 5.2, we demonstrate the advantage of distilling rationales over conventional distillation.

4 Experiments

This section outlines the datasets, baselines, implementation details, and evaluation metrics employed in our experiments and the main results.

4.1 Backbone Models

We employ LLaMA-2-7B-Chat (Touvron et al., 2023), Mistral-7B-Instruct-v0.2 (Jiang et al., 2023) and Qwen2.5-7B-Instruct (Yang et al., 2024a) as the backbone models in this work.

4.2 Datasets

Our experiments involve fine-tuning LLMs using general machine translation data and evaluating their performance across both translation and broader capabilities, including instruction following, safety alignment, and mathematical reasoning. The datasets and benchmarks used for fine-tuning and evaluation are listed below:

General Machine Translation. For parallel training data, we adopt the human-written data collected by Xu et al. (2024). This data comprises human written test datasets from WMT'17 to WMT'20, plus the development and test sets from Flores-200 (Goyal et al., 2022). It covers 4 English-centric language pairs, considering both from and to English directions: Czech (`cs`), Chinese (`zh`), German (`de`), and Russian (`ru`). The WMT'22 test dataset for the same eight translation directions is used for testing. Translation performance is evaluated using the COMET metric (`Unbabel/wmt22-comet-da`) (Rei et al., 2022).

Conversation and Instruction Following. MT-Bench (Zheng et al., 2023) and AlpacaEval (Dubois et al., 2024) are employed to evaluate the conversation and instruction-following abilities of the models. MT-Bench consists of a set of challenging multi-turn questions across various categories. GPT-4 is utilized as the judge, as outlined by Zheng et al. (2023). The AlpacaEval and AlpacaEval 2.0 leaderboard evaluates the models on 805 prompts from the AlpacaEval dataset and calculates the win rate against text-davinci-003 and GPT-4-1106. For this evaluation, we use the `weighted_alpaca_eval_gpt4_turbo` annotator as the judge.

Safety Alignment. Safety is evaluated using harmful behavior datasets consisting of unsafe prompts. Following WalledEval (Gupta et al., 2024), we feed 520 unsafe prompts from AdvBench (Zou et al., 2023) into the LLMs and utilize `LLaMA-3-Guard-8B` (Dubey et al., 2024) to assess whether the responses are harmful. We report the safe rate, defined as the percentage of safe responses across all prompts.

Math reasoning. The reasoning ability is evaluated using GSM8K (Cobbe et al., 2021), which comprises 8.8k high-quality arithmetic word problems designed at the grade school level, to assess the arithmetic reasoning abilities of LLMs. The evaluations are conducted using `lm-evaluation-harness` (Gao et al., 2024).

4.3 Baselines

We compare RaDis with the following fine-tuning approaches:

- **Vanilla Fine-tuning** directly fine-tunes the backbone model using translation data;
- **Multi-task** fine-tunes the LLM with both translation and open-sourced instruction following data.
- **Seq-KD** (Kim and Rush, 2016; Khayrallah et al., 2018) employs sequence-level knowledge distillation along with fine-tuning to alleviate forgetting;
- **SDFT** (Yang et al., 2024b) leverages the backbone LLM to paraphrase the original training data and fine-tunes the model using the synthesized data;

Please refer to Appendix A for further details of the baselines.

4.4 Training Details

Given the constraints of our computational resources, the Low-Rank Adaptation (LoRA) technique (Hu et al., 2022) is utilized in most of our experiments. Specifically, a LoRA adapter with a rank of 16 is integrated into all the linear layers of the LLMs and exclusively trains the adapter. The LLMs are fine-tuned for three epochs on the translation dataset, with a learning rate of 1×10^{-4} and a cosine annealing schedule. The batch is set to 128 for stable training. Our implementation is based on `LLaMA-Factory` (Zheng et al., 2024b). After the fine-tuning phase, the LoRA module is merged into the backbone LLM for testing. For further details, please refer to Appendix C.

Table 1: The performance on machine translation, instruction following, safety, and reasoning benchmarks. Translation performance is averaged across all 4 languages tested. The best and second best results are marked in **Bold** and underlined, respectively.

Models	Machine Translation		Conversation and Instruction Following			Safety	Reasoning
	X→EN	EN→X	MT-bench	AlpacaEval	AlpacaEval 2.0	AdvBench	GSM8K
Backbone LLM: LLaMA-2-7B-chat							
Backbone	79.01	73.39	6.51	71.40	9.66	100.00	21.83
w/ Vanilla-FT	**81.71**	**82.81**	1.5	2.18	0.71	37.88	4.32
w/ Multi-task	81.75	82.59	5.64	44.55	3.98	98.65	11.98
w/ Seq-KD	78.37	73.28	**6.59**	67.48	**8.33**	**100.00**	19.48
w/ SDFT	80.79	75.48	5.66	67.55	7.09	98.08	**20.02**
w/ RaDis (Ours)	81.22	82.61	6.56	**67.94**	7.47	**100.00**	19.48
Backbone LLM: Mistral-7B-Instruct-v0.2							
Backbone	80.84	67.79	7.67	84.91	15.09	68.46	41.62
w/ Vanilla-FT	**82.33**	84.31	1.94	6.07	1.02	4.23	0.23
w/ Multi-task	82.09	**84.44**	6.87	49.46	5.45	**63.85**	22.97
w/ Seq-KD	80.91	74.84	6.99	**82.06**	**12.7**	60.58	**41.77**
w/ SDFT	79.43	52.17	7.00	78.32	10.02	48.27	41.09
w/ RaDis (Ours)	81.94	84.39	**7.57**	80.34	11.05	62.12	41.70
Backbone LLM: Qwen2.5-7B-Instruct							
Backbone	80.85	82.92	8.58	88.46	31.55	99.81	87.72
w/ Vanilla-FT	83.45	85.34	3.83	4.87	2.95	27.12	28.35
w/ Multi-task	**83.90**	**85.55**	7.67	50.58	9.14	98.85	72.1
w/ Seq-KD	80.93	84.01	**8.58**	**87.64**	26.9	**99.62**	88.02
w/ SDFT	81.45	85.36	7.48	67.7	18.91	93.85	83.09
w/ RaDis (Ours)	83.66	85.43	8.43	85.62	**27.91**	99.42	**88.32**

4.5 Results

The performance on machine translation and general abilities, including instruction following, safety, and reasoning, is shown in Table 1.

Fine-tuning is a double-edged sword. In the EN→X direction, Vanilla-FT significantly enhances translation performance compared to zero-shot results, achieving an average COMET score improvement of **+9.45**. In the X→EN direction, the performance improvement is relatively small (**+2.26** COMET). This is mainly because the backbone LLMs already have a strong ability to translate to English. However, this improvement in translation proficiency comes at the cost of a substantial decline in general capabilities, as reflected by the sharp performance drop in instruction-following, safety, and reasoning benchmarks.

RaDis balances translation proficiency and general abilities. Multi-task achieves performance comparable to Vanilla-FT. However, its performance on general tasks declines significantly despite the inclusion of additional instruction-following data. This is because the external instruction data is of low quality and out-of-distribution relative to the backbone LLM. As a result, fine-tuning these data does not alleviate the issue of catastrophic forgetting. The translation results of SDFT fall below the performance of vanilla fine-tuning. Additionally, SDFT shows weaker performance in EN→X translations compared to X→EN. This under-performance stems from the fact that the prompt used for rewriting data is sensitive to the model and task, which does not generalize well. Seq-KD excels in preserving general capabilities but brings almost no improvement in translation performance since it suffers from severe optimization conflict (Section 5.2). In contrast, RaDis strikes a better balance between translation proficiency and general ability. It achieves a COMET score comparable to Vanilla-FT while preserving up to 92.50% of the general capabilities.

5 Analysis

5.1 RaDis Preserves the Advantage of LLMs in Machine Translation

Previous studies primarily evaluate translation quality using general machine translation tasks. However, such evaluations may fail to capture the nuanced capabilities of large language models (LLMs) in translation, particularly their ability to follow instructions. The capacity to adhere to diverse and complex user instructions is crucial for generating translations aligned with user

Table 2: The result of formality steering in translation. The best performance is marked in **Bold**.

	Formal		Informal	
	COMET	Acc	COMET	Acc
LLaMA-2	72.22	74.0	67.08	91.4
w/ Vanilla-FT	**80.16**	82.9	78.79	26.0
w/ RaDis	79.61	**87.3**	**78.80**	**72.4**
Mistral-v0.2	62.23	91.9	58.62	73.9
w/ Vanilla-FT	80.46	82.4	**79.25**	25.9
w/ RaDis	**80.69**	**83.0**	79.11	**46.3**
Qwen2.5	80.86	81.1	79.1	98.7
w/ Vanilla-FT	82.39	81.9	**81.52**	53.9
w/ RaDis	**82.80**	**94.3**	81.31	**87.0**

preferences. To assess this aspect, we introduce instruction-following translation tasks. Specifically, we utilize CoCoA-MT (Nadejde et al., 2022), which evaluates a model's ability to control formality in translation. This dataset consists of 600 English source sentences, each paired with both formal and informal reference translations. Using natural language prompts, we instruct the LLM to generate either formal or informal translations. Following Stap et al. (2024), we set German as the target language and report both formality steering accuracy and overall translation performance. As shown in Table 2, LLMs demonstrate an ability to follow formality instructions, though their translation quality remains relatively poor. While fine-tuning improves translation performance, it diminishes the models' ability to adhere to instructions, often resulting in unintended translations. In contrast, RaDis maintains strong instruction adherence while achieving comparable translation quality, a crucial factor for effective user interaction.

5.2 RaDis Avoids the Conflict between Learning and Mitigating CF

As demonstrated in Section 3.3, our proposed RaDis can be viewed as a specialized form of sequence-level distillation, where the rationale $\mathbf{r}$ serves as the distillation target. However, while both methods excel at preserving general capabilities, RaDis notably enhances translation proficiency, whereas Seq-KD does not. We posit that the difference arises from whether the regularization term conflicts with the MT learning process. In Seq-KD (Equation 5), the MT loss $-\log P(\mathbf{y}|\mathbf{x}, \mathcal{I})$ and the regularization term $-\log P(\mathbf{y}'|\mathbf{x}, \mathcal{I})$ share the same input but have different outputs, which may lead to conflict in optimization. In contrast, with RaDis (Equation 4), the MT loss and the regularization term $-\log P(\mathbf{r}|\mathbf{y}, \mathbf{x}, \mathcal{I})$ are less likely to exhibit this issue.

We analyze gradient similarity to validate our assumption. Specifically, we sample 128 examples from each translation direction to create a validation set consisting of 1024 samples. Subsequently, the gradient features for the MT loss and the regularization term for both methods are extracted, following Xia et al. (2024). Finally, the cosine similarity of the gradient features is computed. As depicted in Figure 5, in 24 out of 32 layers, the gradient of the regularization term for Seq-KD exhibits negative similarity with the MT loss, indicating a significant conflict between these objectives. In contrast, in 25 out of 32 layers, the gradient of RaDis' regularization term shows positive similarity with the MT loss, suggesting that RaDis can avoid the conflict between learning and mitigating forgetting.

5.3 Which is the Key: Rationale Quality or Self-distillation Property?

The effectiveness of RaDis can be explained in two ways: rationale quality (in terms of the knowledge they contain) and the self-distillation property. We conducted the following ablation experiment to analyze the impact of these two factors. Specifically, the self-generated rationales in RaDis were replaced by rationales generated by different models, namely LLaMA-2-7B-Chat and LLaMA-3-70B-Instruct (Dubey et al., 2024). Due to the difference in parameter size and fine-tuning data, these models can provide rationales with varying levels of quality, but all lack the self-distillation property. Therefore, it is possible to separate the rationale quality and the self-distillation property to analyze their contributions.

As shown in Table 3, RaDis consistently mitigates the forgetting of general capabilities, regardless of the type of rationales used. When comparing rationales generated by LLaMA-2-7B-Chat and LLaMA-3-70B-Instruct, the latter demonstrates superior performance in both MT and general tasks. However, self-generated rationales demonstrate the strongest ability to retain general capabilities, even outperforming those generated by LLaMA-3-70B-Instruct, highlighting the importance of the self-distillation property. Overall, these ablation results demonstrate that using the model itself as the teacher is the most effective approach, which aligns with the findings presented in Ren et al. (2024).

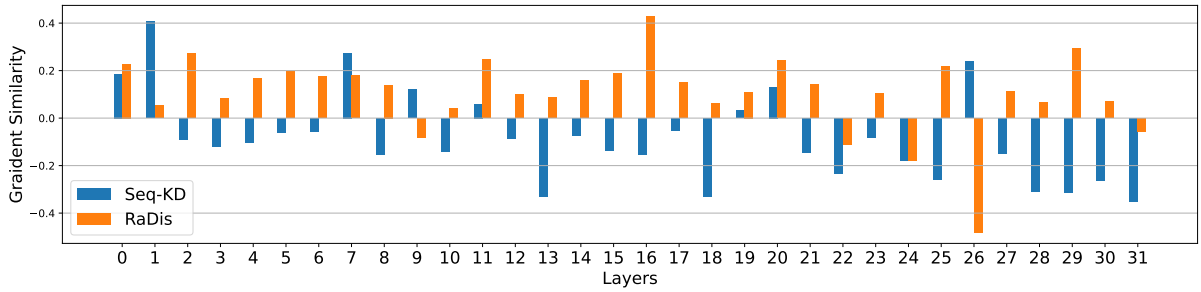

Figure 5: Overview of the gradient similarity between the regularization term and MT loss.

Table 3: The result of ablation study. The names in brackets are the models used to generate rationales. The best results in different RaDis variants are highlighted in **bold**.

Models	Machine Translation		Conversation and Instruction Following			Safety	Reasoning
	X→EN	EN→X	MT-bench	AlpacaEval	AlpacaEval 2.0	AdvBench	GSM8K
Mistral-7B-Instruct-v0.2	80.84	67.79	7.67	84.91	15.09	68.46	41.62
w/ Vanilla-FT	82.33	84.31	1.94	6.07	1.02	4.23	0.23
w/ RaDis (Self-generated)	81.94	84.39	**7.57**	**80.34**	**11.05**	62.12	**41.70**
w/ RaDis (LLaMA-2-Chat-7B)	82.33	84.51	6.89	63.85	6.28	**98.46**	35.03
w/ RaDis (LLaMA-3-70B-Instruct)	**82.84**	**84.71**	7.00	77.41	10.27	58.27	39.42

5.4 Ablation on Rationale Proportion

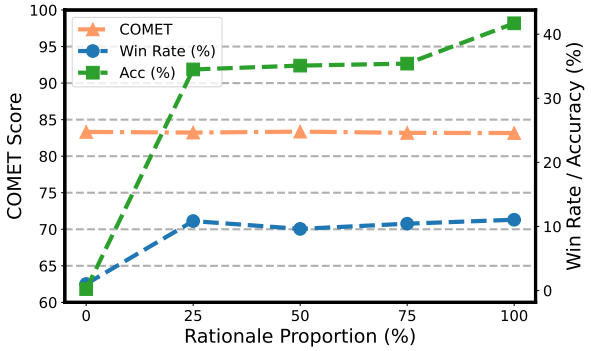

Figure 6: Translation (COMET), instruction following (Win Rate), and math reasoning (Acc) performance with varying proportions of rationales.

To further investigate the impact of rationale proportion, we conducted ablation experiments by combining vanilla fine-tuning with RaDis, varying the proportion of training examples that include rationales. As shown in Figure 6, RaDis achieves the best overall performance on general tasks when rationales are provided for the entire training dataset. However, incorporating rationales in just 25% of the training data significantly mitigates the forgetting of instruction-following and safety tasks. This finding aligns with concurrent works (Zheng et al., 2025; Chen et al., 2025) suggesting that the forgetting of superficial alignment is also superficial and can be easily mitigated. In contrast, the forgetting of more intricate capabilities, such as reasoning, requires a greater proportion of rationales.

5.5 Generalizing to Other Tasks

While we predominately grounded RaDis to the MT task in this paper, RaDis can serve as a universal CIT method for broader tasks. To demonstrate this potential, we conducted experiments on code generation tasks. The results demonstrate that RaDis generalizes well to broader tasks and has the potential to serve as a universal CIT approach. Due to space limits, we refer the readers to Appendix D for further details.

6 Conclusion

In this paper, we demonstrate that LLMs suffer from severe degeneration of general abilities while fine-tuning for translation tasks. To address this issue, we propose a simple yet effective strategy, RaDis. RaDis prompts LLMs to generate rationales for the reference translation and utilizes these rationales to mitigate forgetting in a self-distillation manner. Mirroring the human learning process, these rationales connect prior knowledge with new tasks and tie internal concepts together, thereby enhancing knowledge retention. Extensive experiments show that RaDis greatly enhances the translation performance while preserving the models' general ability, which also benefits translation tasks. Additionally, RaDis provides a fresh angle for utilizing rationales in the CL field and can help future research on building LLMs capable of excelling in specialized tasks without compromising their generality or safety.

Limitations

Our study is subject to certain limitations. Owing to constraints in computational resources, we adopt LoRA on models with 7B parameters. Further investigations involving larger models and full fine-tuning remain to be explored. Besides, as a post-training method, RaDis is limited by the language proficiency of backbone LLM. This limits its performance on low-resource language. However, we believe the rapidly evolving multilingual LLMs would narrow this gap. Furthermore, we predominately focus on fine-tuning with machine translation data, applying RaDis to other NLP tasks will further support its effectiveness. This potential direction is what we intend to explore in future work.

Ethical Considerations

This work is dedicated to the field of fine-tuning LLMs for MT. It proposed to use self-generated rationales to aid vanilla fine-tuning and mitigate forgetting of general abilities of LLMs. It helps alleviate the safety risk when fine-tuning downstream tasks. In our experiments, we used publicly available datasets widely employed in prior research, containing no sensitive information to the best of our knowledge. The authors have followed ACL ethical guidelines, and the application of this work poses no apparent ethical risks.

Reproducibility Statement

Codes and model weights will be made public after review to advocate future research. For synthesizing data, we provide several examples in Appendix F. For evaluation, we primarily use greedy decoding to ensure reproducibility, except where specific generation configurations are mandated by certain benchmark tools. Note that evaluations on instruction-following abilities (AlpacaEval and MT-Bench) rely on OpenAI's API. The randomness of API responses may have little impact on the reproducibility of these benchmarks.

References

Duarte M. Alves, José Pombal, Nuno Miguel Guerreiro, Pedro Henrique Martins, João Alves, M. Amin Farajian, Ben Peters, Ricardo Rei, Patrick Fernandes, Sweta Agrawal, Pierre Colombo, José G. C. de Souza, and André F. T. Martins. 2024. Tower: An open multilingual large language model for translation-related tasks. *CoRR*, abs/2402.17733.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Joshua Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating large language models trained on code. *CoRR*, abs/2107.03374.

Runjin Chen, Gabriel Jacob Perin, Xuxi Chen, Xilun Chen, Yan Han, Nina S. T. Hirata, Junyuan Hong, and Bhavya Kailkhura. 2025. Extracting and understanding the superficial knowledge in alignment. *Preprint*, arXiv:2502.04602.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. Free dolly: Introducing the world's first truly open instruction-tuned llm.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar,

Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B. Hashimoto. 2024. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *CoRR*, abs/2404.04475.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2024. A framework for few-shot language model evaluation.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2022. The flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Trans. Assoc. Comput. Linguistics*, 10:522–538.

Shoutao Guo, Shaolei Zhang, Zhengrui Ma, Min Zhang, and Yang Feng. 2024. Sillm: Large language models for simultaneous machine translation. *CoRR*, abs/2402.13036.

Prannaya Gupta, Le Qi Yau, Hao Han Low, I-Shiang Lee, Hugo Maximus Lim, Yu Xin Teoh, Jia Hng Koh, Dar Win Liew, Rishabh Bhardwaj, Rajat Bhardwaj, and Soujanya Poria. 2024. Walledeval: A comprehensive safety evaluation toolkit for large language models. *CoRR*, abs/2408.03837.

Yongquan He, Xuancheng Huang, Minghao Tang, Lingxun Meng, Xiang Li, Wei Lin, Wenyuan Zhang, and Yifu Gao. 2024. Don't half-listen: Capturing key-part information in continual instruction tuning. *CoRR*, abs/2403.10056.

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.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *CoRR*, abs/2310.06825.

Wenxiang Jiao, Jen-tse Huang, Wenxuan Wang, Zhiwei He, Tian Liang, Xing Wang, Shuming Shi, and Zhaopeng Tu. 2023. Parrot: Translating during chat using large language models tuned with human translation and feedback. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 15009–15020. Association for Computational Linguistics.

Zixuan Ke and Bing Liu. 2022. Continual learning of natural language processing tasks: A survey. *arXiv preprint arXiv:2211.12701*.

Huda Khayrallah, Brian Thompson, Kevin Duh, and Philipp Koehn. 2018. Regularized training objective for continued training for domain adaptation in neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation, NMT@ACL 2018, Melbourne, Australia, July 20, 2018*, pages 36–44. Association for Computational Linguistics.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 1317–1327. The Association for Computational Linguistics.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles, SOSP 2023, Koblenz, Germany, October 23-26, 2023*, pages 611–626. ACM.

Yinquan Lu, Wenhao Zhu, Lei Li, Yu Qiao, and Fei Yuan. 2024. Llamax: Scaling linguistic horizons of LLM by enhancing translation capabilities beyond 100 languages. In *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, pages 10748–10772. Association for Computational Linguistics.

Michael McCloskey and Neal J Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24.

Jisoo Mok, Jaeyoung Do, Sungjin Lee, Tara Taghavi, Seunghak Yu, and Sungroh Yoon. 2023. Large-scale lifelong learning of in-context instructions and how to tackle it. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 12573–12589. Association for Computational Linguistics.

Maria Nadejde, Anna Currey, Benjamin Hsu, Xing Niu, Marcello Federico, and Georgiana Dinu. 2022. Cocoa-mt: A dataset and benchmark for contrastive controlled MT with application to formality. In *Findings of the Association for Computational Linguistics:*

NAACL 2022, Seattle, WA, United States, July 10-15, 2022, pages 616–632. Association for Computational Linguistics.

Ricardo Rei, José G. C. de Souza, Duarte M. Alves, Chrysoula Zerva, Ana C. Farinha, Taisiya Glushkova, Alon Lavie, Luísa Coheur, and André F. T. Martins. 2022. COMET-22: unbabel-ist 2022 submission for the metrics shared task. In *Proceedings of the Seventh Conference on Machine Translation, WMT 2022, Abu Dhabi, United Arab Emirates (Hybrid), December 7-8, 2022*, pages 578–585. Association for Computational Linguistics.

Xuan Ren, Biao Wu, and Lingqiao Liu. 2024. I learn better if you speak my language: Understanding the superior performance of fine-tuning large language models with llm-generated responses. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 10225–10245. Association for Computational Linguistics.

Thomas Scialom, Tuhin Chakrabarty, and Smaranda Muresan. 2022. Fine-tuned language models are continual learners. 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 6107–6122. Association for Computational Linguistics.

Haizhou Shi, Zihao Xu, Hengyi Wang, Weiyi Qin, Wenyuan Wang, Yibin Wang, and Hao Wang. 2024. Continual learning of large language models: A comprehensive survey. *CoRR*, abs/2404.16789.

Yuncheng Song, Liang Ding, Changtong Zan, and Shujian Huang. 2025. Self-evolution knowledge distillation for llm-based machine translation. In *Proceedings of the 31st International Conference on Computational Linguistics, COLING 2025, Abu Dhabi, UAE, January 19-24, 2025*, pages 10298–10308. Association for Computational Linguistics.

David Stap, Eva Hasler, Bill Byrne, Christof Monz, and Ke Tran. 2024. The fine-tuning paradox: Boosting translation quality without sacrificing LLM abilities. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 6189–6206. Association for Computational Linguistics.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Yifan Wang, Yafei Liu, Chufan Shi, Haoling Li, Chen Chen, Haonan Lu, and Yujiu Yang. 2024. Inscl: A data-efficient continual learning paradigm for fine-tuning large language models with instructions. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 663–677. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. 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*.

Yuxiang Wei, Zhe Wang, Jiawei Liu, Yifeng Ding, and Lingming Zhang. 2024. Magicoder: Empowering code generation with oss-instruct. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Tongtong Wu, Linhao Luo, Yuan-Fang Li, Shirui Pan, Thuy-Trang Vu, and Gholamreza Haffari. 2024. Continual learning for large language models: A survey. *CoRR*, abs/2402.01364.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. 2024. LESS: selecting influential data for targeted instruction tuning. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Haoran Xu, Young Jin Kim, Amr Sharaf, and Hany Hassan Awadalla. 2024. A paradigm shift in machine translation: Boosting translation performance of large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2024a. Qwen2.5 technical report. *CoRR*, abs/2412.15115.

Wen Yang, Chong Li, Jiajun Zhang, and Chengqing Zong. 2023. Bigtrans: Augmenting large language models with multilingual translation capability over 100 languages. *CoRR*, abs/2305.18098.

Zhaorui Yang, Tianyu Pang, Haozhe Feng, Han Wang, Wei Chen, Minfeng Zhu, and Qian Liu. 2024b. Self-distillation bridges distribution gap in language model fine-tuning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 1028–1043. Association for Computational Linguistics.

Wenpeng Yin, Jia Li, and Caiming Xiong. 2022. Contintin: Continual learning from task instructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 3062–3072. Association for Computational Linguistics.

Donglei Yu, Yang Zhao, Jie Zhu, Yangyifan Xu, Yu Zhou, and Chengqing Zong. 2025. Simulpl: Aligning human preferences in simultaneous machine translation. *arXiv preprint arXiv:2502.00634*.

Shaolei Zhang, Qingkai Fang, Zhuocheng Zhang, Zhengrui Ma, Yan Zhou, Langlin Huang, Mengyu Bu, Shangtong Gui, Yunji Chen, Xilin Chen, and Yang Feng. 2023. Bayling: Bridging cross-lingual alignment and instruction following through interactive translation for large language models. *CoRR*, abs/2306.10968.

Jiawei Zheng, Hanghai Hong, Xiaoli Wang, Jingsong Su, Yonggui Liang, and Shikai Wu. 2024a. Fine-tuning large language models for domain-specific machine translation. *CoRR*, abs/2402.15061.

Junhao Zheng, Xidi Cai, Shengjie Qiu, and Qianli Ma. 2025. Spurious forgetting in continual learning of language models. *Preprint*, arXiv:2501.13453.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, and Yongqiang Ma. 2024b. Llamafactory: Unified efficient fine-tuning of 100+ language models. *CoRR*, abs/2403.13372.

Wenhao Zhu, Hongyi Liu, Qingxiu Dong, Jingjing Xu, Shujian Huang, Lingpeng Kong, Jiajun Chen, and Lei Li. 2024. Multilingual machine translation with large language models: Empirical results and analysis. In *Findings of the Association for Computational Linguistics: NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 2765–2781. Association for Computational Linguistics.

Andy Zou, Zifan Wang, J. Zico Kolter, and Matt Fredrikson. 2023. Universal and transferable adversarial attacks on aligned language models. *CoRR*, abs/2307.15043.

A Baseline details

Vanilla Fine-tuning. This method directly fine-tunes the backbone LLMs with translation data without incorporating any mechanism to address the forgetting issue.

Sequence-level Knowledge Distillation (Seq-KD). (Kim and Rush, 2016; Khayrallah et al., 2018) This method first sends the formatted translation instructions to the backbone LLM and generates outputs $\mathbf{y}'$. The model is trained with both golden references $\mathbf{y}$ and the self-generated outputs $\mathbf{y}'$. The overall training objective is:

$$\begin{aligned}\mathcal{L}_{\text{Seq-KD}} =& \mathcal{L}(\mathbf{x}, \mathbf{y}) + \mathcal{L}(\mathbf{x}, \mathbf{y}') \\ =& -\log P(\mathbf{y}|\mathbf{x}, \mathcal{I}) - \log P(\mathbf{y}'|\mathbf{x}, \mathcal{I})\end{aligned} \tag{5}$$

Self-distillation Fine-tuning (SDFT). (Yang et al., 2024b) This method first prompts the backbone LLM to paraphrase the original responses present in the task dataset, yielding a distilled dataset. Subsequently, the distilled dataset, which is used in subsequent fine-tuning, helps narrow the distribution gap between LLM and the original dataset. We adopt the general distillation template provided in their paper to paraphrase the dataset.

Multi-task fine-tuning (Multi-task) . This method employs open-sourced instruction following datasets and fine-tunes the LLM with both translation and instruction following data. Specifically, we adopt Alpaca (Taori et al., 2023) and Dolly (Conover et al., 2023) as the chosen instruction following dataset. Note that multi-task fine-tuning utilizes more data in the training process and is usually considered the upper bound of the continual learning approaches.

B Full Results of Machine Translation

The detailed results for general machine translation are shown in Table 4 and Table 5.

C Training Details

C.1 Prompt Templates

In all experiments, we use the original instruction format of the backbone LLM for both rationale generation and fine-tuning. For LLaMA To avoid the overfit on specific instructions. 5 different translation instructions are generated and randomly applied to each sample. The instructions are shown in Figure 7. The prompts used in formality steering translation are shown in Figure 8.

C.2 Hyperparameter

Due to the limitation of resources, our experiments utilize the Low-Rank Adaptation (LoRA) technique (Hu et al., 2022). Specifically, we integrate a LoRA adapter with a rank of 16 into all the linear layers of the LLMs and exclusively train the adapter. The LLMs are fine-tuned over three epochs on the translation dataset, which equates to approximately 2,500 steps. We use a learning rate of 1×10^{-4} and a batch size of 128 to ensure stable training across most experiments. An exception is Seq-KD, which requires a batch size of 256 to maintain the same number of training steps. All experiments are performed on 4 NVIDIA A100 80GB GPUs. For data synthesis, we employ vllm (Kwon et al., 2023) to facilitate fast data generation. For evaluation, we primarily use greedy decoding to ensure reproducibility, except where specific generation configurations are mandated by certain benchmark tools.

D Comparison with Open-source LLM-based MT Models

To compare with open-source LLM-based MT models, We shifted the translation test set to the WMT'23 test set. We report the performance of the best model in our paper (Qwen2.5+RaDis) alongside ALMA and TowerInstruct.

As shown in Table 6, Qwen2.5+RaDis consistently outperforms TowerInstruct-v0.2 in terms of preserving general abilities. This is primarily because TowerInstruct-v0.2 is fine-tuned using UltraChat, which, like other open-sourced instruction datasets, suffers from lower quality.

In terms of translation, TowerInstruct-v0.2 achieves higher performance, largely due to the benefits of multilingual pre-training and extensive parallel fine-tuning. However, we would like to emphasize the strong potential of our approach from two key perspectives:

- **RaDis is more efficient**: The training times for TowerBase 7B and 13B were 80 and 160 GPU days, respectively, using A100-80GB GPUs. Fine-tuning TowerInstruct adds an additional 200 GPU hours. In contrast, RaDis requires only 20 GPU hours (4 hours for generating rationales and 16 hours for training), which is less than 1% of the training cost for TowerInstruct-7B, while still achieving strong performance.

- **RaDis can benefit from stronger backbone LLM**: While TowerInstruct achieves better translation performance, RaDis can effectively bridge this gap by leveraging a stronger backbone LLM. As shown in 'Table 2', switching the backbone from Mistral to Qwen2.5 leads to substantial improvements across all tasks and outperforms ALMA. We believe that as open-source multilingual LLMs continue to improve, the performance gap in translation will gradually narrow.

Together, these results underscore the advantages of our approach and demonstrate that RaDis offers a novel and competitive paradigm for building LLMs that excel in both translation proficiency and general ability.

E Generalizing RaDis to other tasks

In this paper, we predominately grounded RaDis to the MT task. However, RaDis can serve as a universal CIT method for broader tasks. In this section, we demonstrate this potential with the code generation task. Specifically, we fine-tuned Mistral-v0.2 on Python code data from the Magicoder dataset (Wei et al., 2024) and evaluated its performance using HumanEval (Chen et al., 2021) and general ability benchmarks.

As shown in Table 7, RaDis outperforms Vanilla-FT and SDFT in code generation tasks, achieving higher Pass@1 on HumanEval and excelling in other benchmarks for general abilities.

A key reason for this is that RaDis directly preserves the original references in the dataset, whereas SDFT paraphrases them. Intuitively, while paraphrasing helps bridge the distribution gap, it also reduces the amount of learnable knowledge.

Table 4: The overall translation performance (COMET score) in X→EN.The delta performance compared to the backbone LLM is shown.

Models	Czech	German	Russian	Chinese	Avg.
Backbone LLM: LLaMA-2-7B-chat					
Backbone LLM	79.53	81.20	80.36	74.95	79.01
w/ Vanilla-FT	82.74	83.31	82.70	78.08	81.71
w/ Multi-task	82.71	83.37	82.73	78.20	81.75
w/ Seq-KD	78.50	80.61	79.88	74.48	78.37
w/ SDFT	81.93	82.60	81.87	76.77	80.79
w/ RaDis (Ours)	81.75	83.07	82.22	77.83	81.22
Backbone LLM: Mistral-7B-Instruct-v0.2					
Backbone LLM	81.88	81.73	81.97	77.76	80.84
w/ Vanilla-FT	83.51	83.35	83.23	79.21	82.33
w/ Multi-task	82.84	83.04	83.15	79.31	82.09
w/ Seq-KD	81.66	81.98	82.50	77.49	80.91
w/ SDFT	80.38	79.72	80.21	77.39	79.43
w/ RaDis (Ours)	82.41	82.86	83.32	79.17	81.94
Backbone LLM: Qwen2.5-7B-Instruct					
Backbone LLM	81.09	80.73	81.48	80.11	80.85
w/ Vanilla-FT	84.19	83.83	84.62	81.19	83.45
w/ Multi-task	84.87	84.20	85.06	81.45	83.90
w/ Seq-KD	79.17	82.01	82.41	80.14	80.93
w/ SDFT	81.49	80.89	82.41	81.02	81.45
w/ RaDis (Ours)	84.44	83.93	84.82	81.46	83.66

As a result, SDFT may struggle to outperform Vanilla-FT on certain tasks. In contrast, RaDis directly utilizes the original references, preserving all the knowledge embedded in the data.

Regarding performance on general tasks, RaDis still outperforms SDFT. We believe this can be attributed to the distribution gap. While SDFT claims to distill the dataset, it paraphrases the data. As a result, the model's responses are sampled from the paraphrased instruction's output distribution, which tends to be out-of-distribution relative to the original task instruction. In contrast, RaDis performs self-distillation using rationales, which are fully in-distribution. This enables RaDis to more effectively alleviate forgetting and better preserve general abilities.

These results suggest that RaDis generalizes well to a broader range of tasks, highlighting its potential as a robust, general-purpose continual instruction tuning method. We plan to investigate this potential in future works.

F Rationale Examples

Several examples of rationales generated by Mistral-7B-Instruct-v0.2 are provided in Figure 10,11,12,13,14,15,16.

Table 5: The overall translation performance (COMET score) in EN→X. The delta performance compared to the backbone LLM is shown.

Models	Czech	German	Russian	Chinese	Avg.
Backbone LLM: LLaMA-2-7B-Chat					
Backbone LLM	70.14	75.10	75.76	72.57	73.39
w/ Vanilla-FT	81.80	82.81	84.67	81.96	82.81
w/ Multi-task	81.67	82.58	84.24	81.86	82.59
w/ Seq-KD	70.17	74.40	75.62	72.93	73.28
w/ SDFT	68.59	75.21	79.67	78.45	75.48
w/ RaDis (Ours)	81.77	82.39	84.31	81.98	82.61
Backbone LLM: Mistral-7B-Instruct-v0.2					
Backbone LLM	67.39	67.87	64.56	71.32	67.79
w/ Vanilla-FT	84.33	83.04	86.23	83.63	84.31
w/ Multi-task	84.79	82.64	86.47	83.87	84.44
w/ Seq-KD	74.30	73.69	73.10	78.28	74.84
w/ SDFT	51.90	53.32	47.07	56.38	52.17
w/ RaDis (Ours)	84.32	82.95	86.55	83.75	84.39
Backbone LLM: Qwen2.5-7B-Instruct					
Backbone LLM	79.05	81.27	84.77	86.59	82.92
w/ Vanilla-FT	82.47	83.74	88.07	87.08	85.34
w/ Multi-task	82.18	83.96	88.67	87.41	85.55
w/ Seq-KD	80.95	82.06	85.79	87.24	84.01
w/ SDFT	82.86	83.57	87.92	87.08	85.36
w/ RaDis (Ours)	82.33	83.67	88.54	87.32	85.43

Table 6: Comparison to open-sourced LLM-based MT models. The best result in each column is marked in **bold**. The second best is *italicized*.

Models	Machine Translation		Conversation and Instruction Following			Safety	Reasoning
	X→EN	EN→X	MT-bench	AlpacaEval	AlpacaEval 2.0	AdvBench	GSM8K
Qwen2.5-7B-Instruct	80.90	80.50	**8.58**	**88.46**	**31.55**	**99.81**	*87.72*
w/ RaDis	*82.13*	*82.81*	*8.43*	*85.62*	*27.91*	*99.42*	**88.32**
ALMA-7B	81.65	81.91	2.80	1.08	0.17	-	0.00
TowerInstruct-7B-v0.2	**82.77**	**84.28**	5.71	51.59	4.02	30.19	7.35

Instruction 1:
Could you please translate this sentence from {lang1} to {lang2}?
{sent1}
Instruction 2:
Translate the following sentence from {lang1} to {lang2}:
{sent1}
Instruction 3:
Translate this sentence from {lang1} to {lang2}.
{sent1}
Instruction 4:
Translate from {lang1} to {lang2}:
{sent1}
Instruction 5:
{sent1}
Translate this sentence to {lang2}.

Figure 7: The translation instructions.

Formal Translation:
Please translate the following sentence from English to German in a formal tone: {sentence}
Informal Translation:
Please translate the following sentence from English to German in a informal tone: {sentence}

Figure 8: The formality steering translation instructions.

Table 7: Experiments on code generation. The best result in each column is marked in **bold**.

Models	Code Generation	Conversation and Instruction Following		Safety	Reasoning
	HumanEval	AlpacaEval	AlpacaEval 2.0	AdvBench	GSM8K
Mistral-7B-Instruct-v0.2	36.59	84.91	15.09	68.46	41.62
w/ Vanilla-FT	42.07	73.89	8.75	40.00	**43.97**
w/ SDFT	40.24	78.58	10.46	48.08	40.86
w/ RaDis	**43.90**	**80.25**	**11.4**	**51.92**	42.91

Instruction:
Translate this sentence from Czech to English.
Policie celý případ řádně prošetřila a útočníka posléze vypátrala.

Response:
The police investigated the case thoroughly and finally tracked down the assailant.

Rationale:
Explanation:
In this sentence, "policie" means "police," "celý případ" means "the entire case," "řádně prošetřila" means "thoroughly investigated," and "vypátrala" means "tracked down" or "found." The past tense verbs "prošetřila" and "vypátrala" both end in "-la," which is the feminine singular past tense ending in Czech. Since there is no definite article before "policie," it is assumed to be in the neutral form, which is why the verb forms are in the feminine singular. In English, we keep the same meaning but change the word order to "the police investigated the case thoroughly and finally found the assailant."

Figure 9: Example of rationale containing word/phrase translation.

Instruction:
Could you please translate this sentence from Czech to English?
Tak to vypadá, když se lidé za odpočinkem a zábavou přesunou někam, kde je to velmi odlišné od jejich každodenního života.

Response:
This is when people go to a place that is very different from their regular day-to-day life to relax and have fun.

Rationale:
or
It looks like this when people move to a place that is quite different from their everyday life for relaxation and enjoyment.

Figure 10: Example of rationale containing alternative translation.

Instruction:
Translate the following sentence from English to Chinese:
But denuclearization negotiations have stalled.

Response:
但无核化谈判陷入了僵局。

Rationale:
This translation is provided for information purposes only and may not be 100% accurate. It is always recommended to consult a professional translator for important documents or conversations.

Figure 11: Example of rationale containing helpful&safety information.

Instruction:
Translate from English to Czech:
It will support their training.

Response:
Podpoří jejich výcvik.

Rationale:
This sentence means that the action or thing being referred to will help or contribute to the training of the people mentioned.

Figure 12: Example of rationale containing semantic explanation.

Instruction:
Could you please translate this sentence from English to Chinese?
According to their research, a single tea bag can release more than 11.6 billion plastic particles, "which is higher than the plastic load of other earlier foods by several magnitudes".

Response:
根据这项研究显示，他们发现，单一茶袋释出超过116亿颗塑料微粒，“较先前其他食物的塑料负荷量高出好几个级数”。

Rationale:
This sentence translates to:
According to their research, a single tea bag releases over 11.6 billion plastic particles, which is several magnitudes higher than the plastic load of other earlier foods.

Figure 13: Example of rationale containing back-translation.

Instruction:
Translate from English to Czech:
Its second claw was larger, giving rise to the name Hesperonychus which means "western claw."

Response:
Jeho druhý dráp byl větší, a proto byl pojmenován Hesperonychus, což znamená „západní dráp“.

Rationale:
This dinosaur was a theropod, a group of dinosaurs that includes meat-eating species. Hesperonychus lived during the Late Cretaceous period, around 75 to 73 million years ago, in what is now the southwestern United States. Its fossils have been found in the Morrison Formation in Colorado and Utah.
Hesperonychus was a relatively small theropod, with an estimated length of about 3 meters (10 feet) and weight of around 30 kilograms (66 pounds). It had a long, low snout and small, sharp teeth, suggesting that it may have been an agile predator that hunted small prey.
The most distinctive feature of Hesperonychus was its second toe, which was much larger than the others. This feature gave rise to its name, as "hespero" means "western" and "nychus" means "claw." The large claw was likely used for gripping and holding onto branches or prey, making Hesperonychus a unique and intriguing dinosaur.
Despite its small size, Hesperonychus was an important discovery in the study of dinosaur evolution, as it provided evidence of the diversity of theropod foot structures during the Late Cretaceous period.

Figure 14: Example of rationale containing factual supplement.

Instruction:
Could you please translate this sentence from Czech to English?
„I kdyby všechno ostatní na světě bylo špatně, on byl nebroušený diamant, který ozářil každý den“, řekl."

Response:
"With everything else that’s going wrong with the world, he was that diamond in the rough that was shining bright every day," he said.

Rationale:
This idiomatic expression "diamond in the rough" refers to someone who has great potential or value, despite not being polished or perfect yet.

Figure 15: Example of rationale containing semantic explanation.

Instruction:
Wir können froh sein, dass wir diese Zeit längst überwunden haben.
Translate this sentence to English.

Response:
We can be glad that we have long overcome this time.

Rationale:
Explanation: The sentence in German means "We can be glad that we have got over this time a long time ago." The past perfect tense "haben überwunden" (have overcome) is used to express that an action was completed in the past. The subjunctive mood "können sein" (can be) is used to express a state of being or a condition. The past participle "überwunden" agrees with the subject "wir" (we) in gender and number. The article "diese" refers to the time mentioned earlier in the sentence. The word "längst" (long) is an adverb that modifies the verb "überwunden" and indicates that the action was completed a long time ago. The sentence in English conveys the same meaning, but uses the past perfect continuous tense "have been glad" to express the same idea of a past state that continues up to the present.

Figure 16: Example of rationale containing grammar information.