Challenge Track: LoRAs in All Directions - Directional Adapters and Noisy-Channel Reranking for Indic MT

Sajay Raj
Woxsen University
sajayraj08@gmail.com

Abstract

Low-resource machine translation for Indic languages remains challenging, especially when high-resource languages such as Hindi and English must be translated to and from very low-resource, grammatically rich languages like Bhili, Mundari, Santali, and Gondi.

We describe our winning system for a recent shared task in this setting. We start from a strong pretrained Indic MT backbone, IndicTrans2, and fine-tune it jointly on all translation directions, pushing the model close to memorization under strict data constraints. On top of this backbone, we add direction-specific low-rank adapters based on LoRA that allow each language pair to specialize while still sharing most parameters. At inference time, we further couple these directional adapters through a noisy-channel objective, in which forward and reverse models jointly score a set of candidate translations, encouraging outputs that are both fluent in the target language and informative about the source.

This combination of shared pretraining, directional parameter-efficient adaptation, and noisy-channel reranking substantially improves over a strong fine-tuned baseline. We release our codebase at https://github.com/SajayR/LoRA-in-All-Directions.

1 Introduction

The MMLoSo shared task (mml, 2025) targets a difficult gap in machine translation: bridging high-resource languages (Hindi, English) with extremely low-resource community languages (Bhili, Mundari, Santali, Gondi). These languages are particularly challenging for standard models because they utilize diverse scripts (such as Ol Chiki) and complex word structures, yet lack the large-scale parallel data required to learn these features effectively.

In this regime, standard training strategies face a dilemma. Training separate models for each direction creates data fragmentation, leading to poor convergence. Conversely, joint multilingual fine-tuning maximizes transfer learning but introduces *interference* (or "negative transfer"), where the model's capacity is dominated by high-resource directions, often resulting in script hallucinations or morphological simplification in the lower-resource targets.

Our winning submission addresses this trade-off through a *saturate-then-specialize* strategy. We hypothesize that while a shared backbone is necessary to learn general linguistic representations, distinct parameter spaces are required to resolve orthographic and grammatical conflicts. We therefore combine massive joint fine-tuning (to saturate the backbone with domain knowledge) with direction-specific Low-Rank Adapters (LoRA) (to isolate task-specific constraints) (Hu et al., 2022). Finally, to counter the semantic drift common in low-resource generation, we abandon greedy decoding in favor of a noisy-channel formulation (Pang et al., 2022), using the LoRA adapters to strictly enforce mutual consistency between source and translation.

2 Task and Data

2.1 Shared task setup

The shared task focuses on translation between two high-resource languages (Hindi, English) and four low-resource Indic languages: Bhili, Mundari, Santali, and Gondi. Let

$$\mathcal{L} = \{\text{Hindi}, \text{English}, \text{Bhili}, \text{Mundari}, \text{Santali}, \text{Gondi}\}$$

be the set of languages.

A translation direction is defined as an ordered pair $d = (\ell_s \rightarrow \ell_t)$ with $\ell_s, \ell_t \in \mathcal{L}$. The task provides parallel datasets $\mathcal{D}_d = \{(x^{(i)}, y^{(i)})\}_{i=1}^{N_d}$ for each direction, where source x and target y are in their native scripts. The objective is to produce a translation $\hat{y}$ given x and the direction d.

Proceedings of MMLoSo 2025: First Workshop on Multimodal Models for Low-Resource Contexts and Social Impact, pages 101–105

2.2 Leaderboard Score

The leaderboard reports a single scalar score S that mixes BLEU(Post, 2018) and chrF(Popović, 2015) across all eight directions.

Let $\text{BLEU}_{H\to L}$ be the mean BLEU over the four high→low directions (Hin→Bhi, Hin→Mun, Hin→Gon, Eng→San), and let $\text{BLEU}_{L\to H}$ be the mean over the four low→high directions (Bhi→Hin, Mun→Hin, Gon→Hin, San→Eng). We define $\text{chrF}_{H\to L}$ and $\text{chrF}_{L\to H}$ analogously, replacing BLEU with chrF.

The final score is

$$S = 0.6\big(0.6\,\text{BLEU}_{H\to L} + 0.4\,\text{BLEU}_{L\to H}\big) \quad (1)$$
$$+\,0.4\big(0.6\,\text{chrF}_{H\to L} + 0.4\,\text{chrF}_{L\to H}\big). \quad (2)$$

BLEU contributes 60% of S and chrF 40%; within each metric, high→low directions get 60% of the weight and low→high directions 40%.

Two variations of this score are reported: the **Public Score**, calculated on a fixed 25% subset of the test data during the competition, and the **Private Score**, calculated on the remaining 75% hidden subset to determine the final rankings.

3 Methodology

4 Backbone Selection

We initially compared **NLLB-200** (600M) (NLLB Team et al., 2022) and **IndicTrans2** (360M) (Gala et al., 2023) by fine-tuning both for 100k steps. Table 1 shows that IndicTrans2 outperformed NLLB by nearly 9 points despite being half the size.

Backbone	Public Score
NLLB-600M	243.18
IndicTrans2-360M	**252.11**

Table 1: Leaderboard scores at 100k steps.

Tokenization analysis (Table 2) reveals that NLLB suffers from higher word fragmentation across the board. In contrast, IndicTrans2 offers far superior tokenization stability. While this comes at the cost of a slightly higher unknown token rate (Table 2), the difference is negligible ($< 1.7\%$ worst-case) and easily mitigated via decoding constraints. We therefore proceed with IndicTrans2.

4.1 Tag-Only Preprocessing

To minimize pipeline complexity and avoid brittle external preprocessors for these under-resourced languages, we adopt a "tags-only" preprocessing strategy. We avoid script unification or transliteration. Instead, we condition the model purely via tag prefixing (Johnson et al., 2017).

	Fertility (↓)		Unknown tokens % (↓)	
Language	**NLLB**	**IndicTrans2**	**NLLB**	**IndicTrans2**
Bhili	1.73	1.45	0.02	0.03
Gondi	2.16	1.75	0.17	0.21
Mundari	2.56	2.16	0.42	0.50
Santali	3.07	1.44	0.00	1.69

Table 2: **Backbone Analysis.** Fertility scores (lower is better) and unknown-token rates (lower is better) for NLLB and IndicTrans2.

Each language $\ell \in \mathcal{L}$ is associated with a fixed tokenizer tag $\tau(\ell)$ (e.g., $\tau(\text{Hindi}) = \texttt{hin\_Deva}$). For extremely low-resource languages not originally supported by the tokenizer, we map them to the closest available script-proxy tag. Specifically, we map Bhili to `mar_Deva` (Marathi) as a surrogate to leverage Devanagari script transfer.

For a source sentence x and direction $d = (\ell_s \to \ell_t)$, we construct the model input

$$\tilde{x} = [\tau(\ell_s), \tau(\ell_t), \text{tokens}(x)].$$

By consistently applying this formatting, we convert all parallel data into a unified sequence-to-sequence task, allowing joint training across all directions simultaneously.

4.2 Base Model and Joint Fine-tuning

We initialize our model with **IndicTrans2**, a Transformer-based encoder-decoder (Vaswani et al., 2017) model pretrained on large-scale Indic corpora. While IndicTrans2 is a strong baseline, the specific domains and languages in this shared task (e.g., Gondi, Mundari) are under-represented in the pretraining data.

We treat the union of all available training data $\mathcal{D} = \bigcup_d \mathcal{D}_d$ as a single dataset. We fold the development sets into the training data to maximize the supervision available for the lowest-resource directions. We fine-tune all model parameters θ (initialized at pretrained weights θ_0) via standard token-level cross-entropy loss:

$$\mathcal{L}_{\text{base}}(\theta) = -\sum_{(x,y,d)\in\mathcal{D}} \frac{1}{|y|}\sum_{t=1}^{|y|} \log p_\theta(y_t \mid y_{<t}, \tilde{x}_d) \quad (3)$$

where $\tilde{x}_d$ encodes the direction d via tags. This stage produces a "generalist" base model $\theta^\star$ that

creates a strong baseline but may suffer from interference between conflicting translation directions (e.g., translating into Devanagari vs. Ol Chiki scripts).

4.3 Directional LoRA Adapters

To mitigate interference and allow specialization, we freeze the base model $\theta^\star$ and introduce direction-specific Low-Rank Adaptation (LoRA) modules.

For a target module weight matrix $W \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ (e.g., attention projections or FFN layers), we parameterize the update for direction d as:

$$W_d = W + \frac{\alpha}{r} B_d A_d \tag{4}$$

where r is the rank, $A_d \in \mathbb{R}^{r \times d_{\text{in}}}$, $B_d \in \mathbb{R}^{d_{\text{out}} \times r}$, and α is a scaling factor.

We create a separate bank of adapters $\{\Delta\theta_d\}$ for each direction. During this stage, we freeze $\theta^\star$ and optimize only the adapter parameters $\Delta\theta_d$ and the shared embeddings/LM head ϕ on the subset of data $\mathcal{D}_d$ corresponding to that direction. This results in a system where the backbone provides shared linguistic knowledge, while the adapter defines the specific mapping for a language pair.

5 Inference: Noisy-Channel Reranking

Standard beam search often yields generic or "safe" translations, particularly in low-resource settings where the model may hallucinate or default to copying the source script. To address this, we employ a noisy-channel reranking approach (Pang et al., 2022) that couples forward and reverse translation models.

5.1 Candidate Generation and Scoring

Given a test input x and direction $d = (\tau_s \to \tau_t)$, we first generate a set of K candidate translations $\mathcal{Y}_K = \{y^{(1)}, \ldots, y^{(K)}\}$ using beam search with the forward adapter $\Delta\theta_d$. We strictly constrain the beam search to disallow the generation of the <unk> token to prevent degenerate outputs in low-resource target scripts.

We then score each candidate $y^{(k)}$ using two components:

1. **Forward Score:** The log-probability of the candidate given the source, using the forward adapter $\Delta\theta_d$:

$$\ell_{\text{fwd}} = \frac{1}{|y^{(k)}|} \log p(y^{(k)} \mid x, d; \Delta\theta_d) \tag{5}$$

2. **Reverse Score:** The log-probability of reconstructing the source x given the candidate, using the reverse adapter $\Delta\theta_{d^{-1}}$ (where $d^{-1} = \tau_t \to \tau_s$):

$$\ell_{\text{rev}} = \frac{1}{|x|} \log p(x \mid y^{(k)}, d^{-1}; \Delta\theta_{d^{-1}}) \tag{6}$$

Both scores are computed via batched teacher forcing.

5.2 Reranking Objective

The final translation $\hat{y}$ is selected by maximizing a weighted combination of these scores:

$$\hat{y} = \underset{y^{(k)} \in \mathcal{Y}_K}{\text{argmax}} \left[\alpha \cdot \ell_{\text{fwd}} + \beta \cdot \ell_{\text{rev}}\right] \tag{7}$$

In our experiments, we set $\alpha = 1.0$ and $\beta = 1.0$. The reverse term acts as a regularizer: it penalizes candidates that are fluent (high forward probability) but semantically drifted such that the source cannot be reconstructed.

Finally, we apply a lightweight script-aware post-processing step to normalize punctuation and remove artifacts (e.g., spacing before Danda or Ol Chiki punctuation) introduced by the tokenizer.

6 Experimental Setup

6.1 Training Details

We trained the base model for 300k steps with a learning rate of 2e-5, using mixed precision (BF16) and a batch size of 44. For the LoRA stage, we used a rank $r = 64$, $\alpha = 128$, and dropout 0.1. We targeted all linear layers in the attention and feed-forward blocks along with training the base model's embedding and output head while training for 50k steps.

7 Results

Table 3 presents the performance of our system on the shared task leaderboard. We compare two model sizes (360M and 1.1B) across three stages of our pipeline: the fine-tuned baseline, the addition of Directional Adapters (MultiLoRA), and the final Noisy-Channel Reranking (Backloss).

Impact of Directional Adapters For the 1.1B model, Directional LoRA adapters give a $+13.9$ point increase in the Public score. This validates our hypothesis that low-resource languages benefit from dedicated parameter spaces that are isolated from the interference of other translation directions. Notably, our 360M model with adapters

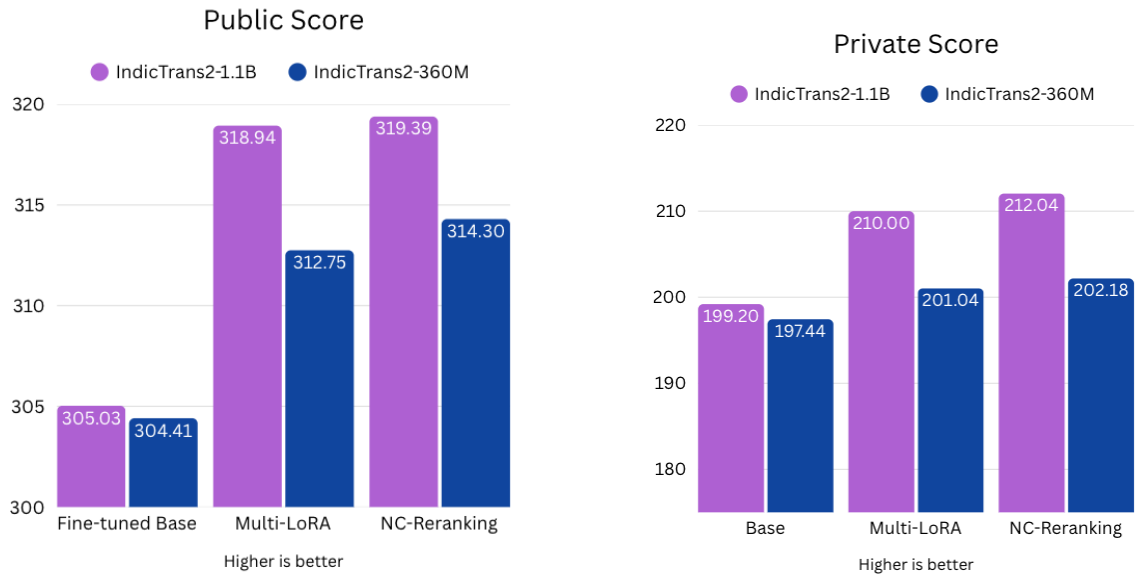

Figure 1: **Final Results** Left: Public Scores (higher is better), showing a steady increase in performance across both model sizes with the 3 stages (Finetuning, Multi-LoRA adaptation, Noisy Channel Reranking). Right: Private Scores. Showing a similar trend in performance gains

Model Configuration	Public	Private
Small Variant (360M)		
IndicTrans2-360M (FT Baseline)	304.41	197.44
+ Directional Adapters	312.75	201.04
+ Noisy-Channel Reranking	314.30	202.18
Large Variant (1.1B)		
IndicTrans2-1.1B (FT Baseline)	305.03	199.20
+ Directional Adapters	318.94	210.00
+ Noisy-Channel Reranking	**319.39**	**212.04**

Table 3: **Main Results.** Comparison of Public and Private leaderboard scores. Our proposed methods yield consistent improvements across model sizes. The 1.1B model with full pipeline achieves the winning score.

(312.75) significantly outperforms the much larger 1.1B baseline (305.03), highlighting the efficiency of this approach.

Impact of Noisy-Channel Reranking The addition of noisy-channel reranking provides a consistent final boost (+0.45 to +1.55 points). While the magnitude is smaller than the LoRA step, this reranking method is a cheap and consistent method to improve MT performance.

8 Conclusion

We presented our winning submission to the MMLoSo shared task. By combining a strong pretrained backbone (IndicTrans2) with a unified "tags-only" preprocessing scheme, we established a robust baseline. We then introduced Directional LoRA Adapters to resolve interference between diverse scripts and Noisy-Channel Reranking to ensure semantic fidelity. Our results demonstrate that even in the era of massive multilingual models, task-specific modular adaptation and rigorous decoding strategies remain essential for achieving state-of-the-art performance in low-resource Indic languages.

9 Limitations

The proposed system remains constrained by the IndicTrans2 tokenizer and vocabulary. Coverage of low-resource scripts (in particular Ol Chiki for Santali and the surrogate tag used for Bhilli) is incomplete, which leads to fragmented subword segmentations and occasional out-of-vocabulary symbols. Decoding-time constraints such as banning <unk> partially mitigate their impact on automatic metrics, but do not recover missing characters and can still yield approximate or distorted surface forms for rare words and named entities.

The training and tuning setup is tightly coupled to the shared-task configuration. The base model is deliberately saturated on the full training data, and several hyperparameters (e.g., beam size, noisy-channel weights) are selected using subsets of the same data or leaderboard feedback, rather than a clean held-out validation set.

Finally, the architecture makes explicit trade-offs in efficiency and generality. Direction-specific LoRA adapters scale linearly with the number of

language pairs and require separate finetuning for each direction, limiting zero-shot coverage. The noisy-channel reranking scheme further increases inference-time cost by requiring both forward and reverse likelihoods for multiple candidates per input, which may be impractical in latency- or resource-constrained settings.

Acknowledgments

I am deeply grateful to my parents for their constant encouragement and for generously supporting the compute resources required for this work. I also thank the MMLoSo shared task organizers for designing the task, providing the data, and maintaining the evaluation infrastructure.

References

2025. Multimodal models for low-resource contexts and social impact 2025 language challenge: Shared task on translation for low-resource indic languages. Kaggle Competition. Co-located with the MMLoSo Workshop @ IJCNLP-AACL 2025.

Jigar Gala, Gowtham Ramesh, Sumanth Doddapaneni, and 1 others. 2023. Indictrans2: Towards high-quality and accessible machine translation for indic languages. *arXiv preprint arXiv:2305.16307*.

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 *Proceedings of ICLR*.

Melvin Johnson, Mike Schuster, Quoc V. Le, and 1 others. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

NLLB Team, Marta R. Costa-jussà, James Cross, and 1 others. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Richard Yuanzhe Pang, He He, and Kyunghyun Cho. 2022. Amortized noisy channel neural machine translation. In *Proceedings of the International Natural Language Generation Conference*.

Maja Popović. 2015. chrf: Character n-gram f-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*.