TRepLiNa: Layer-wise CKA+REPINA Alignment Improves Low-Resource Machine Translation in Aya-23 8B

Toshiki Nakai[1], **Ravi Kiran Chikkala**[1], **Lena Sophie Oberkircher**[1], **Nicholas Jennings**[1],
Natalia Skachkova[2], **Tatiana Anikina**[2], **Jesujoba O. Alabi**[1]

[1]Saarland University [2]German Research Center for Artificial Intelligence (DFKI)
{toshiki3738,lenaoberkircher}@gmail.com
{rach00004@teams,s8nijenn@stud,jalabi@cs}.uni-saarland.de

Abstract

The 2025 Multimodal Models for Low-Resource Contexts and Social Impact (MMLoSo) Language Challenge addresses one of India's most pressing linguistic gaps: the shortage of resources for its diverse low-resource languages (LRLs). The challenge focuses on developing a translation model capable of translating between High resource languages (HRLs) (Hindi/English) and LRLs (Bhili, Mundari, Santali, and Gondi). In this study, we use the MMLoSo 2025 challenge dataset to investigate whether enforcing cross-lingual similarity in specific internal layers of a decoder-only multilingual large language model (LLM) can improve translation quality from LRLs to HRLs. Specifically, we combine Centered Kernel Alignment (CKA), a similarity metric that encourages representations of different languages to align with Representation Projection Invariance (REPINA), a regularization method that constrains parameter updates to remain close to the pretrained model, into a joint method, we call TRepLiNa (CKA + REPINA). Our results[1] show that aligning mid-level layers with TRepLiNa is a low-cost and practical way to improve LRL translation in data-scarce settings. We make our code and models public.

1 Introduction

Many multilingual LLMs share parameters across languages, yet transfer to low-resource languages (LRLs) often lags behind their performance on high-resource languages (HRLs) (Conneau et al., 2020; Zhang et al., 2020). Recent analysis of Aya-23 8B (Aryabumi et al., 2024), a multilingual decoder-only model, shows strong neuron overlap across related languages in the embedding layer, perhaps due to token overlap, but it exhibits a marked drop in overlap at intermediate and higher layers (Trinley et al., 2025). This suggests a simple hypothesis: *selectively increasing cross-lingual similarity where it is weakest (mid/high layers) may lead to better transfer for LRLs.* We focus only on the LRL→HRL translation, based on the intuition that models generally find it easier to understand a new language than to generate it (Lin et al., 2025). We operationalize this via a lightweight alignment loss between hidden representations of parallel sentences, which is applied at a chosen layer ℓ. We use centered kernel alignment (CKA) (Kornblith et al., 2019), which can robustly compare representations across networks and layers, together with representation projection invariance (REPINA) (Razdaibiedina et al., 2023) to stabilize HRL features against representation drift. We perform experiments, using zero-shot (Zhao et al., 2023), few-shot (Karimi Mahabadi et al., 2022) and QLoRA-based fine-tuning (Zhang et al., 2023) on Aya-23 8B, using the MMLoSo benchmark (lrl, 2025) pairs, Hindi/English pivots as HRLs; Bhili (Indo-Aryan), Mundari (Austro-asiatic), Santali (Austro-asiatic) and Gondi (Dravidian) as LRLs.

Our work makes the following **contributions:**

- We present, to the best of our knowledge, the first systematic study of *layer-wise* alignment in a decoder-only LLM for low-resource machine translation (MT), comparing CKA and TRepLiNa (CKA+REPINA) across layers.
- We demonstrate that mid-layer alignment (roughly layers 10–15) is most effective, with TRepLiNa consistently favoring layer 15 in limited-data settings.
- We show improvements in the weighted composite score of BLEU (Papineni et al., 2002) and ChrF (Popović, 2015), defined as (0.6×BLEU+0.4×ChrF) with TRepLiNa and provide guidelines on when and where alignment should be applied.

[1] https://github.com/konta3738/cka-repina-aya23

Proceedings of MMLoSo 2025: First Workshop on Multimodal Models for Low-Resource Contexts and Social Impact, pages 25–34
December 23, 2025 ©2025 Association for Computational Linguistics

2 Related Work

Low-Resource Transfer Methods for Indic LRLs: Alongside alignment-based methods, zero-shot and few-shot strategies have also been explored for Indic LRLs. Huidrom and Lepage (2020) show that a single multilingual Neural Machine Translation (NMT) model can translate between unseen Indian language pairs, with performance improving as small amounts of parallel data are added. Ghosal et al. (2025) address the problem of improving few-shot generation for Indic LRLs through prompt refinement for MT and other downstream generation tasks. Their findings highlight the importance of designing techniques that enhance low-resource performance. While they focus on input-level prompting, we complement this by aligning hidden representations across layers to improve transfer for Indic LRLs.

Cross-lingual Alignment Methods: Cross-lingual alignment has long been studied as a way to enhance transfer in multilingual models, particularly for LRLs (Hämmerl et al., 2024). Post-hoc cross-lingual alignment methods rotate representations after training, e.g., SVD/orthogonal Procrustes or projection-based removal of language-specific components, improving zero-shot transfer (Deb et al., 2023; Yang et al., 2021). Joint optimization injects alignment during training, e.g., cosine-similarity objectives on parallel sentences or contrastive InfoNCE setups (Wieting et al., 2019; Pan et al., 2021) while balancing negatives. CKA has emerged as a computationally attractive alternative to Canonical Correlation Analysis (CCA) (Hotelling, 1936) for comparing intermediate activations and for distillation/analysis (Dasgupta and Cohn, 2025). REPINA (Razdaibiedina et al., 2023) regularizes against representation collapse/drift. We apply these ideas to layer-wise alignment in Aya-23 8B for LRL MT.[2]

3 Data

In this research project, we use the MMLoSo shared task train dataset (lrl, 2025) for the experiments with roughly 20k sentence pairs per direction, splitting the dataset into 95% train and 5% development. The language pairs include Bhili↔Hindi, Mundari↔Hindi, Gondi↔Hindi (all in Devanagari script) and Santali↔English (Santali in Ol Chiki script and English in Roman script). Our initial tokenization analysis of the data shows that Santali has higher tokenization fertility. It often requires a longer maximum sequence length (368) than Hindi/English (256), which can slightly reduce the tokenwise parallelism available to the alignment loss when sequences must be truncated to apply CKA.

4 Methodology and Experiments

In our experiments, we focus on Aya-23 8B, a strong openly available model with broad typological coverage and robust multilingual capabilities. The model is pretrained on 23 languages, including Hindi and English, but it does not cover Mundari, Bhili, Gondi, or Santali. We issue all prompts instructions in English.

4.1 Prompting

Here, we discuss the zero-shot and few-shot prompting methods that are used in the experiments.

Zero-shot: In zero-shot experiments, the model relies on its knowledge without any examples (Chikkala et al., 2025). We consider zero-shot as the baseline for the experiments. See Figure 3 for zero-shot prompt template in the Appendix.

Few-shot: In few-shot experiments, we use examples for each language pair from the train set as reference for the language model (Anikina et al., 2025). For each language pair, we use the first example from the training split of the provided data for one-shot, the first three for three-shot, and the first five for five-shot. See Figure 4 for few-shot prompt template in the Appendix.

4.2 TRepLiNa

This section describes the alignment objective of TRepLiNa. Figure 1 illustrates an overview of our proposed training method.

Given a parallel pair $(x^{(A)}, x^{(B)})$ from an LRL A and a pivot HRL B, let $H_\ell^{(A)}, H_\ell^{(B)} \in \mathbb{R}^{T\times d}$ denote token wise hidden states at layer ℓ (sequence length T, width d) and $H_{\text{pre}\,\ell}^{(A)}$ be the hidden states obtained from the pretrained model (with an adapter disabled). We augment the MT loss (token-level cross entropy) L_{MT} with (i) a CKA alignment between LRL/HRL representations and

[2] We focus on CKA here; exploring cosine/contrastive or newer similarity objectives (e.g., Listopad 2025) is left to future work.

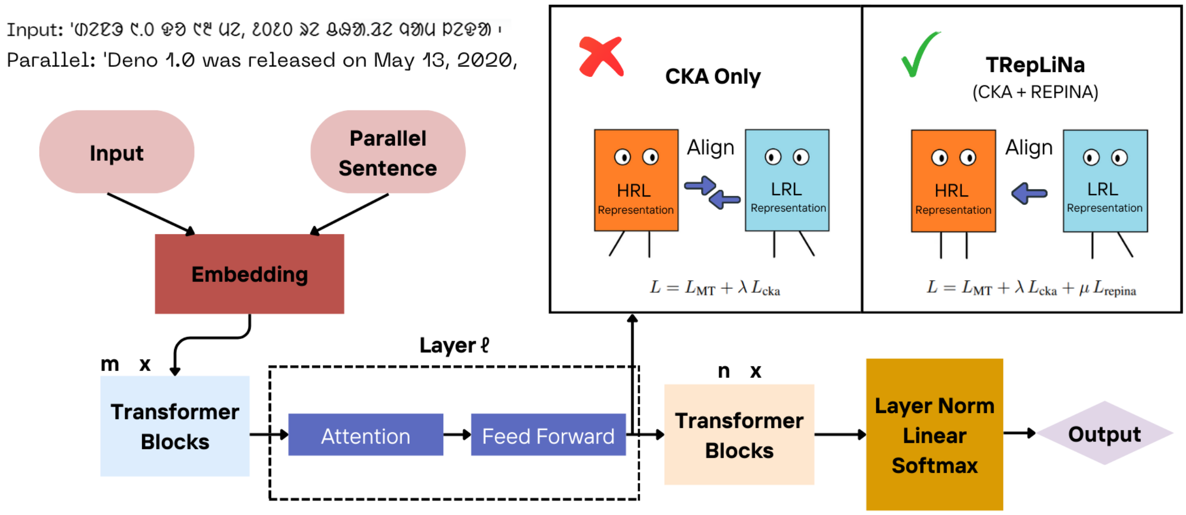


Figure 1: Proposed alignment architecture. Under **CKA-only**, both HRL and LRL representations drift toward each other, potentially distorting HRL features. By contrast, **TRepLiNa** constrains HRL representations while guiding LRL representations toward them, achieving targeted alignment without degrading HRL quality. Here, m and n denote the number of transformer blocks before and after the target alignment layer, respectively.

(ii) a REPINA anchoring term that resists drift of HRL features:

$$L = L_{\mathrm{MT}} + \lambda\, L_{\mathrm{CKA}} + \mu\, L_{\mathrm{REPINA}} \tag{1}$$

with $\lambda, \mu > 0$. We use linear CKA on mean–centered features:

$$L_{\mathrm{CKA}} = 1 - \mathrm{CKA}(H_\ell^{(A)}, H_\ell^{(B)}),$$
$$\mathrm{CKA}(H_\ell^{(A)}, H_\ell^{(B)}) = \frac{\|X^\top Y\|_F^2}{\sqrt{\|X^\top X\|_F^2}\,\sqrt{\|Y^\top Y\|_F^2}}. \tag{2}$$

F denotes Frobenius norm. X and Y represent the matrices after applying mean-centering on $H_\ell^{(A)}$ and $H_\ell^{(B)}$ respectively. For REPINA, we anchor HRL states to a stop-gradient identity mapping of a reference pass, i.e.,

$$L_{\mathrm{REPINA}}(H_{\mathrm{pre}\,\ell}^{(A)}, H_\ell^{(A)}) = \| H_{\mathrm{pre}\,\ell}^{(A)} - \tilde{\phi}(H_\ell^{(A)}) \|_2^2, \tag{3}$$

Equivalently, $\tilde{\phi}(\cdot) = \mathrm{sg}(\cdot)$; in our implementation this is the detached HRL hidden state at the same layer from the forward pass. CKA pulls A toward B at layer ℓ, while REPINA stabilizes B. Unless noted, both terms are applied at a single layer ℓ.

4.3 Experimental Design

Step 1: Layer sweep (small data): To make the sweep computationally tractable, we sample 1,000 parallel pairs and train for one epoch per direction (Mundari → Hindi, Santali → English). We sweep layers $\ell \in \{1, 2, 5, 10, 15, 20, 25, 30, 31, 32\}$ and evaluate **CKA-only** and **TRepLiNa (CKA+REPINA)** against two baselines **NoAlign** and **REPINA-only**. For **REPINA-only**, we fix $\ell = 15$ (the best layer observed under TRepLiNa) to isolate the marginal contribution of CKA. We set $\lambda = \mu = 0.05$, values that are large enough to reveal effects at this data scale, yet small enough to avoid the over-alignment; larger CKA weights (e.g., $\lambda = 0.3$) degraded MT performance in preliminary runs. The **NoAlign** (standard QLoRA finetuning) excludes both CKA and REPINA.

Step 2: Longer training at the best layer: Using the best layer from Step 1, we train for up to 5 epochs and track BLEU/ChrF on a 500-sample development set each epoch, comparing TRepLiNa vs. REPINA-only ($\lambda = 0.01$, $\mu = 0.05$).

5 Results and Analysis

Here, we analyze the results of zero-shot, few-shot, TRepLiNa, REPINA and NoAlign from Table 1

5.1 Step 1: Layer-Wise Trends

The result is discussed for 1k pairs and 1 epoch. For Mundari–Hindi, the weighted composite score across layers improves (see Figure 2). CKA peaks at layer 10, whereas TRepLiNa peaks at layer 15; the same tendency holds for Santali–English (see Appendix B.1).

Interpretation: CKA-only encourages both languages to meet in the middle; without a stabilizer, HRL features may drift, which can blunt gains in

Language	Zeroshot	Few-shot (1)	Few-shot (3)	Few-shot (5)	TRepLiNa (Ours)	REPINA-only	NoAlign
Bhili→Hindi	4.75	4.54	4.84	3.96	47.96	**48.02**	48.01
Gondi→Hindi	4.39	3.66	3.75	3.99	**36.26**	36.18	36.25
Mundari→Hindi	3.54	3.00	3.01	3.24	**34.24**	33.45	33.36
Santali→English	1.38	1.77	1.05	1.16	**33.02**	32.28	32.14

Table 1: Final translation scores across language pairs (0.6 × BLEU + 0.4 × ChrF). Best scores are in **bold.**

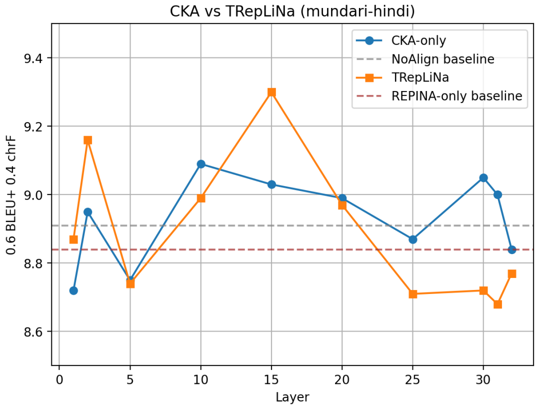


Figure 2: Comparison of (0.6× BLEU +0.4× ChrF) across layers for CKA, REPINA, NoAlign and TRepLiNa.

later layers. REPINA counteracts this, making mid-high layers (15) the sweet spot when pairing with CKA.

5.2 Step 2: Multi-Epoch Comparison at Selected Layer

Setup: Using the best alignment layer from Step 1 (typically a mid-layer around $\ell = 15$), we train for up to five epochs on the full split (≈20k pairs) and evaluate after each epoch on a 500-sample development set. Unless noted otherwise, we set $(\lambda, \mu) = (0.01, 0.05)$ for this longer run, i.e., a lower CKA weight than in Step 1 to avoid over-regularization at scale. We report the MMLoSo score (0.6×BLEU + 0.4×ChrF) and also track BLEU/ChrF separately (Appendix Table 2). Model selection uses the best development set checkpoint per direction.

5.3 Findings

Gondi→Hindi: TRepLiNa attains the highest performance score exceeding zero-shot performance. Few-shot(1) has the lowest score, the gap between the highest and lowest performance scores is 32.6.

Mundari→Hindi: TRepLiNa achieves the best score on development set outperforming zero-shot, while few-shot(1) has the lowest score, the difference between the best and the lowest performance score is 29.24.

Santali→English: TRepLiNa has the best performance score surpassing zero-shot, whereas Few-shot(3) has the lowest score. A difference of 31.97 exists between the best and worst performance scores. For comparison Billah et al. (2024) report a BLEU of 11.13 on their development set; our result (Appendix Table 2) is 25.24 BLEU, a +14.11 absolute and ≈2.27× relative improvement.

Bhili→Hindi: REPINA-only has the highest score, it could be because Bhili and Hindi are typologically close, a strong CKA weight can over-align and wash out beneficial language-specific features. However, our approach TRepLiNa performs better than zero-shot. Few-shot(5) has the lowest score, The highest score exceeds the lowest by 44.06.

Takeaways: (i) *Early vs. late epochs:* **NoAlign** shows stronger performance in the initial stages of training with 1k inputs (see Figure 2), whereas **REPINA-only** tends to surpass it when trained on larger datasets (20k). (ii) *Data scaling:* Larger datasets favor a lower CKA weight; we used $\lambda = 0.05$ for the 1k/1-epoch sweep and $\lambda = 0.01$ for 20k/5-epoch training. As cross-lingual representations become sufficiently aligned, excessive CKA pressure can erode language-specific cues. (iii) *Language proximity:* For related pairs (e.g., Bhili–Hindi), We recommend reducing λ; for more distant pairs, mid-layer TRepLiNa remains robust.

6 Conclusions

In this paper, we investigate layer-wise alignment as a simple and effective strategy for improving low-resource translation using Aya-23 8B on MMLoSo language pairs. We show that aligning representations at mid layers enhances performance on translation tasks between language pairs, and that coupling similarity (CKA) with stability (REPINA) in our proposed **TRepLiNa** method yields robust gains across data-scarce settings.

Limitations

We do not explore other similarity objectives (cosine, contrastive InfoNCE) or recent proposals (Listopad, 2025); we use coefficients (λ, μ) without scheduler/tuning; and this study does not include a thorough ablation study of the hyperparameters (λ, μ). In our experiments, we have not explored chain of thought prompting techniques and different prompt templates. From the results Table 1, we observe that there is a reduced performance of TRepLiNa on Bhili→Hindi, where it underperforms the REPINA-only and NoAlign methods. These results indicate that our method may not generalize well to all language pairs. Santali tokenization sometimes requires longer sequences than 256, reducing token-wise overlap for alignment when truncation occurs. We do not evaluate human adequacy/fluency or domain transfer and qualitative analysis of the generated output by the models.

Acknowledgments

We are grateful to our collaborators and mentors for valuable discussions on layer-wise analysis and low-resource MT. We especially thank Mina Abarico for designing the architecture diagram. This project was also supported by the German Federal Ministry of Research, Technology and Space (BMFTR) as part of the project TRAILS (01IW24005), and by Saarland University. We also thank the reviewers for their constructive comments. Any remaining errors are our own.

References

2025. Multimodal models for low-resource contexts and social impact 2025. Kaggle Competition.

Tatiana Anikina, Ivan Vykopal, Sebastian Kula, Ravi Kiran Chikkala, Natalia Skachkova, Jing Yang, Veronika Solopova, Vera Schmitt, and Simon Ostermann. 2025. dfkinit2b at checkthat! 2025: Leveraging llms and ensemble of methods for multilingual claim normalization.

Viraat Aryabumi, John Dang, Dwarak Talupuru, Saurabh Dash, David Cairuz, Hangyu Lin, Bharat Venkitesh, Madeline Smith, Kelly Marchisio, Sebastian Ruder, Acyr Locatelli, Julia Kreutzer, Nick Frosst, Phil Blunsom, Marzieh Fadaee, Ahmet Üstün, and Sara Hooker. 2024. Aya 23: Open weight releases to further multilingual progress. *Preprint*, arXiv:2405.15032.

Syed Mohammed Mostaque Billah, Ateya Ahmed Subarna, Sudipta Nandi Sarna, Ahmad Shawkat Wasit, Anika Fariha, Asif Sushmit, and Arig Yousuf Sadeque. 2024. Towards santali linguistic inclusion: Building the first santali-to-english translation model using mt5 transformer and data augmentation. *Preprint*, arXiv:2411.19726.

Ravi Kiran Chikkala, Tatiana Anikina, Natalia Skachkova, Ivan Vykopal, Rodrigo Agerri, and Josef van Genabith. 2025. Automatic fact-checking in english and telugu. *ArXiv*, abs/2509.26415.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Sayantan Dasgupta and Trevor Cohn. 2025. Improving language model distillation through hidden state matching. In *The Thirteenth International Conference on Learning Representations*.

Ujan Deb, Ridayesh Parab, and Preethi Jyothi. 2023. Zero-shot cross-lingual transfer with learned projections using unlabeled target-language data. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 449–457, Toronto, Canada. Association for Computational Linguistics.

Soumya Suvra Ghosal, Soumyabrata Pal, Koyel Mukherjee, and Dinesh Manocha. 2025. PromptRefine: Enhancing few-shot performance on low-resource Indic languages with example selection from related example banks. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 351–365, Albuquerque, New Mexico. Association for Computational Linguistics.

Katharina Hämmerl, Jindřich Libovický, and Alexander Fraser. 2024. Understanding cross-lingual Alignment—A survey. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 10922–10943, Bangkok, Thailand. Association for Computational Linguistics.

Harold Hotelling. 1936. Relations between two sets of variates. *Biometrika*, 28(3/4):321–377.

Rudali Huidrom and Yves Lepage. 2020. Zero-shot translation among Indian languages. In *Proceedings of the 3rd Workshop on Technologies for MT of Low Resource Languages*, pages 47–54, Suzhou, China. Association for Computational Linguistics.

Rabeeh Karimi Mahabadi, Luke Zettlemoyer, James Henderson, Lambert Mathias, Marzieh Saeidi, Veselin Stoyanov, and Majid Yazdani. 2022. Prompt-free and efficient few-shot learning with language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3638–3652, Dublin, Ireland. Association for Computational Linguistics.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. 2019. Similarity of neural network representations revisited. *Preprint*, arXiv:1905.00414.

Peiqin Lin, Marion Thaler, Daniela Goschala, Amir Hossein Kargaran, Yihong Liu, André F. T. Martins, and Hinrich Schütze. 2025. Construction-based reduction of translationese for low-resource languages: A pilot study on Bavarian. In *Proceedings of the 7th Workshop on Research in Computational Linguistic Typology and Multilingual NLP*, pages 114–121, Vienna, Austria. Association for Computational Linguistics.

Aleksandr Listopad. 2025. Wave-based semantic memory with resonance-based retrieval: A phase-aware alternative to vector embedding stores. *Preprint*, arXiv:2509.09691.

Xiao Pan, Mingxuan Wang, Liwei Wu, and Lei Li. 2021. Contrastive learning for many-to-many multilingual neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 244–258, Online. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

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, Lisbon, Portugal. Association for Computational Linguistics.

Anastasia Razdaibiedina, Ashish Khetan, Zohar Karnin, Daniel Khashabi, and Vivek Madan. 2023. Representation projection invariance mitigates representation collapse. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 14638–14664, Singapore. Association for Computational Linguistics.

Katharina Trinley, Toshiki Nakai, Tatiana Anikina, and Tanja Baeumel. 2025. What language(s) does aya-23 think in? how multilinguality affects internal language representations. *Preprint*, arXiv:2507.20279.

John Wieting, Taylor Berg-Kirkpatrick, Kevin Gimpel, and Graham Neubig. 2019. Beyond BLEU: Training neural machine translation with semantic similarity. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4344–4355, Florence, Italy. Association for Computational Linguistics.

Ziyi Yang, Yinfei Yang, Daniel Cer, and Eric Darve. 2021. A simple and effective method to eliminate the self language bias in multilingual representations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5825–5832, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020. Improving massively multilingual neural machine translation and zero-shot translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1628–1639, Online. Association for Computational Linguistics.

Xuan Zhang, Navid Rajabi, Kevin Duh, and Philipp Koehn. 2023. Machine translation with large language models: Prompting, few-shot learning, and fine-tuning with QLoRA. In *Proceedings of the Eighth Conference on Machine Translation*, pages 468–481, Singapore. Association for Computational Linguistics.

Xuandong Zhao, Siqi Ouyang, Zhiguo Yu, Ming Wu, and Lei Li. 2023. Pre-trained language models can be fully zero-shot learners. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15590–15606, Toronto, Canada. Association for Computational Linguistics.

A Appendix: Training and Implementation Details

A.1 Codebase and Reproducibility

We provide a single-script trainer for QLoRA fine-tuning of Aya-23 with layer-wise alignment. Seeds are fixed for Python and PyTorch (CPU/GPU). All console/file logs are timestamped; training/eval logs are written via helper functions (`write_train_log`, `write_eval_log`). LoRA adapters are pushed to a Hugging Face repo using access tokens from environment variables.

A.2 Model, Quantization, and LoRA

We load `CohereLabs/aya-23-8B` with 4-bit NF4 (BitsAndBytes) and `bf16` (or `fp16`). We enable `output_hidden_states` to obtain intermediate activations. LoRA is applied to standard projection modules `[q,k,v,o,gate,up,down]` with default $(r{=}16, \alpha{=}32, \text{dropout}{=}0.05)$. We use gradient checkpointing and `enable_input_require_grads()` to support k-bit training.

A.3 Tokenization and Batching

We use a fast tokenizer; if the pad token is missing, EOS is used as PAD. Causal-LM inputs are left-padded; alignment-only passes are right-padded. Prompts follow: *"Translate to* {`lang_b_name`}*:\n{src}\n"*. Labels mask the prompt with -100. Max lengths are typically 256 (Santali uses 368). We pad to a multiple of 8 for tensor cores. Global batch size is 1 with gradient accumulation (default 16).

A.4 Data Splits and Development Set

From a CSV with columns `src_col/tgt_col`, we create train/development set splits. If =<1k examples, development set =10%; otherwise ≈5% (capped 1k–2k). Development set evaluation uses up to 500 examples per epoch.

A.5 Losses and Layer-wise Alignment (No Equations)

Task loss: We use a label-smoothed causal LM loss with $\epsilon = 0.1$ over valid target tokens.

Alignment passes (procedure only): For a parallel pair from LRL A and HRL B, we:

1. Run *source-only* strings for both languages to collect hidden states at a chosen layer ℓ.
2. Mask pads, align sequence lengths (truncate to maximum), flatten tokens across the batch, and mean-center features.
3. Compute a *similarity score* between A and B at layer ℓ and add its complement as an alignment penalty.

This is the same CKA objective introduced in the main text; we omit formulas here and refer the reader to the Methodology and Experiments section (Section 4).

REPINA anchoring (procedure only): Periodically (e.g., every two optimizer steps) we:

1. Disable adapters to obtain a *reference* HRL representation at layer ℓ on the same inputs.
2. Penalize the mean-squared deviation between current and reference HRL hidden states (stop-gradient on the reference).

This follows the REPINA scheme described in the main text; equations are intentionally omitted here.

Combined objective: Training minimizes task loss + similarity penalty + anchoring penalty with user-set coefficients (`--lambda_cka`, `--mu_repina`). Both terms are applied at a single chosen layer ℓ.

A.6 Optimization and Precision

We use PagedAdamW8bit (or AdamW) with $\beta = (0.9, 0.95)$, weight decay 0.01, linear warmup (ratio default 0.05), and LR in $[1 \times 10^{-4}, 2 \times 10^{-4}]$ (default 2×10^{-4}). Mixed precision uses `torch.amp.autocast` (`bf16/fp16`); for `fp16`, gradients use `GradScaler`. We clip global gradients to 1.0 for `bf16`. Gradients are zeroed with `set_to_none=True`. Optimizer steps occur every `grad_accum` micro-steps.

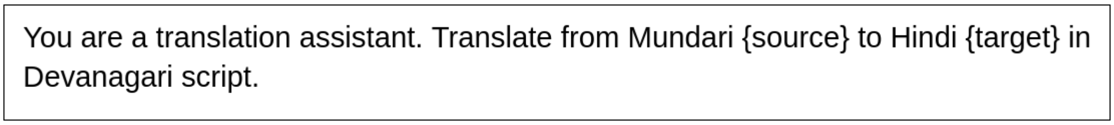
A.7 Model and Training Defaults

Unless noted: max source/target 256 (Santali 368), LR 2×10^{-4}, warmup 5%, batch size 1, grad accumulation 16, and mixed precision. Layer ℓ is selected via sweeps; CKA and REPINA use the same ℓ.

A.8 BLEU and ChrF Results (Per Direction)

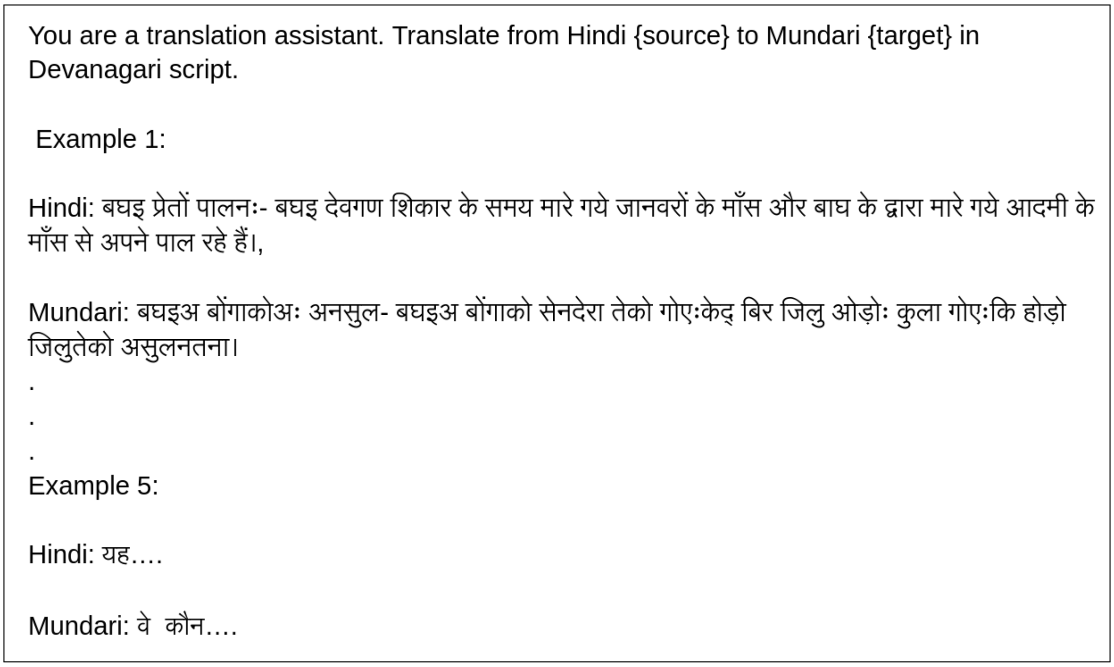
Compute, Runtime, and Practical Notes

- **Hardware:** Experiments are ran on A100 40GB or H100 80GB (QLoRA fits comfort-



You are a translation assistant. Translate from Mundari {source} to Hindi {target} in Devanagari script.

Figure 3: Zero-shot prompt



You are a translation assistant. Translate from Hindi {source} to Mundari {target} in Devanagari script.

Example 1:

Hindi: बघइ प्रेतों पालनः- बघइ देवगण शिकार के समय मारे गये जानवरों के माँस और बाघ के द्वारा मारे गये आदमी के माँस से अपने पाल रहे हैं।,

Mundari: बघइअ बोंगाकोअः अनसुल- बघइअ बोंगाको सेनदेरा तेको गोएःकेद् बिर जिलु ओड़ोः कुला गोएःकि होड़ो जिलुतेको असुलनतना।

.

.

.

Example 5:

Hindi: यह....

Mundari: वे कौन....

Figure 4: Few-shot prompt

ably); BF16 preferred when available. Training took approximately 30 hours on 1 A100 40GB, and 16 hours on 1 H100 80GB.

- **Stability:** For typologically close pairs (e.g., Bhili–Hindi), reduce the similarity weight over epochs to avoid over-alignment.
- **Layer indexing:** Hidden state tuple index 0 corresponds to the embedding output; a user layer ℓ refers to the 1-based transformer block output.

B Appendix B: Complementary Results

B.1 Step-1: Layer Sweep on Santali→English

With only 1,000 training pairs and a single epoch, *anchoring* from REPINA can transiently conflict with task updates: large anchoring (μ) tends to pull parameters back toward the reference HRL representation, partially canceling early task learning. Empirically, λ=0.05, μ=0.05 underperforms **CKA**-only, but reducing anchoring to μ=0.01 makes **TRepLiNa** outperform **CKA**-only. Performance peaks at ℓ=15, suggesting a mid-layer is most effective for aligning Santali to English in

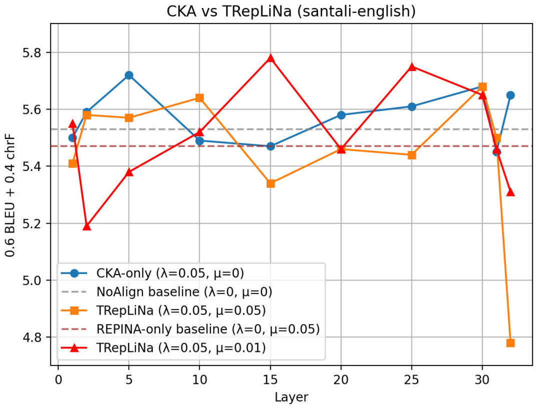


Figure 5: Comparison of (0.6 × BLEU + 0.4 × ChrF) across layers for **CKA** and **TRepLiNa** on Santali→English (1k rows, 1 epoch). Dashed lines indicate each method's baseline.

Language	Zeroshot	Few-shot (1)	Few-shot (3)	Few-shot (5)	TRepLiNa (Ours)	REPINA-only	NoAlign
Bhili→Hindi	0.88	0.64	0.93	0.35	40.15	**40.26**	40.13
Gondi→Hindi	0.37	0.12	0.30	0.56	**28.71**	28.44	28.64
Mundari→Hindi	0.14	0.06	0.04	0.08	**25.94**	25.08	24.93
Santali→English	0.04	0.04	0.03	0.05	**25.24**	24.64	24.26

Table 2: Final translation scores across language pairs (BLEU). Best scores are in **bold.**

Language	Zeroshot	Few-shot (1)	Few-shot (3)	Few-shot (5)	TRepLiNa (Ours)	REPINA-only	NoAlign
Bhili→Hindi	10.57	10.40	10.72	9.38	59.67	59.65	**59.84**
Gondi→Hindi	10.42	8.97	8.93	9.12	47.58	**47.78**	47.67
Mundari→Hindi	8.66	7.43	7.48	7.98	**46.68**	46.02	46.00
Santali→English	3.40	4.39	2.60	2.83	**44.68**	43.74	43.96

Table 3: Final translation scores across language pairs (ChrF). Best scores are in **bold.**

this small-data setting. **Practical note:** for low data/short training, prefer moderate CKA ($\lambda \approx 0.05$) with lighter anchoring ($\mu \approx 0.01$) and sweep mid-layers (e.g., 10–20).

B.2 BLEU Table: Summary and Takeaways

Table 2 compares final BLEU across settings. On **Mundari→Hindi** and **Santali→English**, **TRepLiNa (CKA+REPINA)** achieves the best scores, outperforming both *REPINA-only* and *NoAlign*. For **Bhili→Hindi**, *REPINA-only* narrowly leads. Few-shot and zero-shot remain far below alignment-based methods, indicating that explicit layer-wise alignment is crucial in the low-resource regime.

B.3 ChrF Table: Summary and Takeaways

Table 3 shows the same comparison in ChrF. The pattern largely mirrors BLEU: **TRepLiNa** tops **Mundari→Hindi** and **Santali→English**, while *NoAlign* is slightly best on **Bhili→Hindi**. Despite small differences between top systems on Bhili→Hindi, both metrics agree that alignment generally helps, especially for the more distant pairs. Overall, ChrF confirms the BLEU trends and supports the utility of combining CKA with REPINA.

C Appendix C: Future Directions

Scope: We did not explore HRL → LRL directions in the main paper due to the asymmetric computational profile of the task and the cost of fine-tuning Aya-23 8B. Here we provide a preliminary *Step-1* layer sweep on Hindi→Mundari (1k pairs, 1 epoch; $\lambda = 0.05$, $\mu = 0.05$).

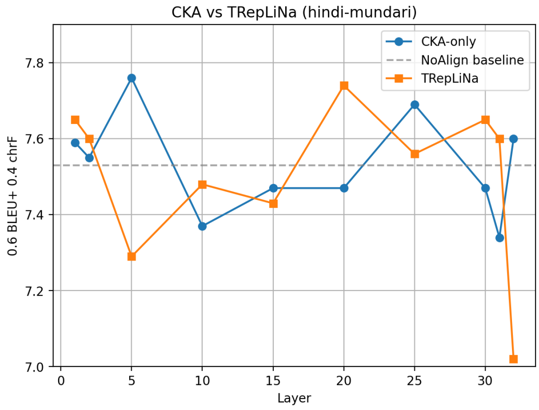


Figure 6: Layer sweep on Hindi→Mundari (1k pairs, 1 epoch). We plot 0.6 × BLEU + 0.4 × ChrF for **CKA-only** and **TRepLiNa**; dashed lines denote each method's NoAlign baseline. CKA-only peaks at ℓ=10, TRepLiNa at ℓ=20.

Setup and metrics: We compare **CKA-only** and **TRepLiNa** against the **NoAlign** baseline across layers, using the combined score $0.6 \times \text{BLEU} + 0.4 \times \text{ChrF}$ (Figure. 6).

Observations: **(i)** CKA-only peaks at layer 10 and TRepLiNa peaks at layer 20; both outperform NoAlign. **(ii)** With $\mu = 0.05$ and such a small regime (1k/1 epoch), REPINA can over-regularize, likely dampening short-term task learning. This suggests TRepLiNa may be more competitive under larger budgets (e.g., 20k/5 epochs), where the auxiliary signal has time to synergize with the task objective.

Layer asymmetry: For LRL → HRL, we observed peaks around layers 10–15 for TRepLiNa, whereas HRL → LRL peaks later (layer 20). One plausible explanation is that Aya-23 8B has limited pretrained support for LRL tokens and structures. When the *output* is an LRL (e.g., Mundari), later layers must adapt themselves to generate unseen languages; when the *input* is an LRL, earlier layers need to map LRL signals into language-agnostic features. We leave a rigorous verification of this hypothesis to future work.

Future work may extend this approach to encoder–decoder or speech–text models, and explore adaptive scheduling strategies for alignment strength in truly low-data scenarios.