

Toward End-to-End Open-Weight LLM-RBMT for Endangered Languages

Jared Coleman¹ Bhaskar Krishnamachari²

¹Loyola Marymount University ²University of Southern California
jared.coleman@lmu.edu bkrishna@usc.edu

Abstract

We present an end-to-end open-weight pipeline for LLM-Assisted Rule-Based Machine Translation (LLM-RBMT) into extremely low-resource languages, with no proprietary-API dependency at any stage and no per-language linguistic effort beyond registering a language with the framework. On Owens Valley Paiute (OVP), a critically endangered Indigenous language of California, we synthesize training data entirely from the OVP language specification, fine-tune two Qwen2.5-3B LoRA adapters (one for each LLM-driven stage of the translation system), and serve both locally. The fine-tuned forward adapter *matches or exceeds* gpt-4o-mini on the 150-sentence forward evaluation of Coleman et al. (2026b) (COMET_c 0.472 vs. 0.448, same evaluator), and a separately fine-tuned backward adapter approaches it on a 40-item hand-validated backward reference set, both running locally. The full recipe runs on a single consumer GPU.

1 Introduction

The LLM-Assisted Rule-Based Machine Translation (LLM-RBMT) paradigm (Coleman et al., 2024, 2026b) combines expert-designed sentence-building tools with large language models (LLMs) to translate English into languages with no publicly available parallel corpora. It has so far relied on closed-weight models (primarily OpenAI’s GPT series) to perform the semantic decomposition of English input into structured representations the rule-based tools can consume. This dependency is a friction for the very communities the approach is designed to serve. Endangered-language revitalization work is often carried out under institutional and ethical arrangements that restrict where linguistic data can be stored or processed, including principles such as CARE (Carroll et al., 2020) and OCAP. Beyond data sovereignty, external API dependence also imposes recurring cost, introduces

unpredictable availability, and limits the community’s ability to customize, adapt, or audit the system over time.

Open-weight models, increasingly capable small LLMs released with their weights, offer a path to sidestep all of these issues. What is unclear is *how much worse* LLM-RBMT approaches perform when the closed-weight model is swapped out for an open-weight, locally-runnable one. In this paper we address this question for Owens Valley Paiute (OVP; ISO 639-3: mnr), a critically endangered Indigenous language of the Owens Valley region of California with fewer than 10 fluent speakers. In Resource Abundance Notation (Coleman et al., 2026a), which summarizes a language’s resource profile as order-of-magnitude counts of speakers, monolingual sentences, and parallel sentences per partner language, OVP is 1/3/en-3 and is among the most resource-scarce profiles for which machine translation has been attempted.

The central idea behind LLM-RBMT is that it never lets the LLM freely generate target-language text (Figure 1). The Yaduha framework realizes each target language as a *language package*: a typed schema of the structures the language can express, together with a closed English-labeled vocabulary of stems, and a deterministic *renderer* that maps a populated schema instance to a well-formed surface sentence by applying the language’s morphology. Translation proceeds in two LLM-driven steps. A **forward translator** takes English input and, under constrained decoding that guarantees a schema-valid output, produces a list of Sentence instances that the renderer converts to OVP. A **backward translator**, also LLM-driven, reads each Sentence instance and produces a natural-English gloss, used both for evaluation and to show the user what the system understood. The model handles meaning while the grammar handles form. Output is grammatical by construction, because only the renderer ever writes OVP.

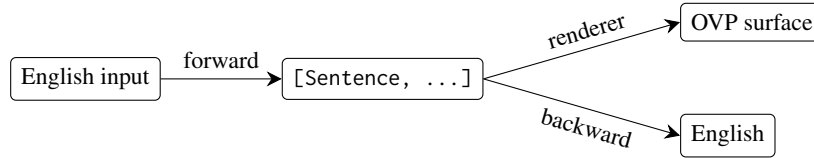


Figure 1: The LLM-RBMT pipeline. The *forward* and *backward* translators are LLM-driven; the *renderer* is deterministic. Either LLM slot can be filled with a proprietary API model or an open-weight model. The structured [Sentence, ...] list is both the rule-based renderer’s input (producing the user-visible OVP surface) and the backward translator’s input (producing the natural-English gloss used for evaluation and for user transparency).

For OVP, the schema comprises two Pydantic Sentence subclasses, **SubjectVerbSentence** and **SubjectVerbObjectSentence**, whose fields cover subject and object (plain noun, agentive-nominalized verb, or pronoun), verb (transitive or intransitive, with tense/aspect), proximity (proximal/distal), and number (singular/dual/plural). A curated vocabulary of 40 nouns and 37 verbs supplies the target-language lexicon. When an input lemma falls outside this vocabulary, the renderer emits a role-tagged placeholder (e.g. [laptop]) so that the output signals its own uncertainty.

A worked example. Consider the input “*They danced.*” The forward translator emits a single schema-valid SubjectVerbSentence:

```
SubjectVerbSentence(
  subject=Pronoun(3.pl),
  verb=IntransitiveVerb("dance", past))
```

The renderer consumes this structure and produces the OVP surface form “*Uhuŵa nüga-ku,*” combining the third-plural pronoun *uhuŵa* with the verb stem *nüga* and the past-tense suffix *-ku*. The backward translator, reading only the structured form (not the surface form, which is unintelligible to a general-purpose English LLM), produces the natural-English rendering “*They danced.*”

When an input contains words that fall outside the curated vocabulary, the structured form carries the original English lemma in the corresponding field. For the input “*Tom caught a fish,*” the forward translator produces:

```
SubjectVerbObjectSentence(
  subject=Noun("Tom"),
  verb=TransitiveVerb("catch", past),
  object=Noun("fish"))
```

whose subject head and verb lemma are **out-of-vocabulary** (OOV) while the object head *fish* is in-vocabulary. The renderer emits “*[Tom]-ii pagwi-noka u-[catch]-ku,*” preserving the inline placeholders while expressing the remaining grammati-

cal content (subject suffix *-ii*, object suffix *-noka*, object-pronoun prefix *u-*, and past-tense suffix *-ku*).

Evaluation. The backward translator plays two roles that are sometimes conflated. It is the primary evaluation instrument, so translation quality is measured by comparing its output to the source English; it is also the user-facing transparency layer, so that when the forward translator cannot express an input exactly, the structured representation and its back-translation let the system say, in effect, “*I cannot say that in OVP, but here is a set of simple sentences that approximates it.*” Because the structured form may still carry OOV English lemmas in its content-word fields, a back-translation can look correct simply because those lemmas were passed through verbatim rather than because the system encoded them grammatically. We therefore distinguish two evaluation arms. In the backwards arm, the backward translator sees the full structured output, OOV lemmas included. In the comparator arm, OOV content-word lemmas are first masked to role-tagged sentinels ([NOUN], [VERB]) before the backward translator reads the structure. The “Tom caught a fish” example with gpt-4o-mini at both LLM positions makes the contrast visible:

Tom caught a fish.

Target	(gpt-4o-mini, forward and backward)
[Tom]-ii pagwi-noka u-[catch]-ku.	
Backwards	
Tom caught that fish.	0.94
Comparator	
The [NOUN] [VERB]ed the fish.	0.37

The numbers are COMET scores that essentially measure the semantic similarity between the input and backward/comparator translations (§4.1). The backwards arm scores near-perfectly because *Tom* and *catch* pass through verbatim; the comparator arm drops sharply because those lemmas are masked, leaving only the in-vocabulary *fish* to be

lexicalized in OVP. The gap between arms measures the extent to which backwards quality is carried by lexical passthrough.

1.1 Contributions

1. An automated procedure for generating a supervised fine-tuning dataset from a Yaduha¹ language specification, requiring no additional linguistic effort from the language-package author. The procedure produces training pairs for both pipeline positions in a single pass, and can itself be driven end-to-end by open-weight models.
2. A fully open-weight LLM-RBMT system for OVP in which both the forward translator and the backward translator are LoRA-fine-tuned Qwen2.5-3B-Instruct adapters trained on data produced by this procedure. The forward adapter matches or exceeds gpt-4o-mini on a 150-sentence OVP forward evaluation set, and the backward adapter approaches it on a 40-item hand-validated backward reference set, while both run locally on a single consumer GPU. The training pipeline (dataset generation and fine-tuning) depends on no proprietary model.
3. The first systematic evaluation of LLM-RBMT with open-weight models substituted for proprietary ones at both the forward and backward translator. Through progressive ablations we separate two senses in which such a pipeline can be “fully open”: that it *runs* with no proprietary API at translation time (open *deployment*), and that the data used to *train* its models was itself produced with no proprietary model (open *training provenance*).

2 Related Work

Low-resource and no-resource MT. Machine translation for low-resource languages is a mature research area, comprehensively surveyed by [Ranathunga et al. \(2023\)](#). Mainstream neural approaches scale impressively across typologically diverse languages ([NLLB Team et al., 2022](#)), but their building blocks (transfer learning from related languages, iterated back-translation, and massively multilingual pre-training) all presuppose at least some naturalistic text in the target language. For critically endangered languages with neither parallel corpora nor substantial monolingual corpora, these techniques do not directly apply. Traditional

rule-based MT frameworks such as Apertium take the opposite approach, relying entirely on hand-written lexicons and transfer grammars. This scales well when community-linguist effort is available but poorly when it is not.

LLM-Assisted Rule-Based MT. [Coleman et al. \(2024\)](#) introduced LLM-RBMT as a hybrid alternative to both extremes: a community-curated grammar specification (modeled as Pydantic Sentence classes in the Yaduha framework) supplies the structural scaffold, and LLMs perform the semantic decomposition that turns natural-language input into structured targets. [Coleman et al. \(2026b\)](#) extended the paradigm with a richer grammar and compared five translator architectures (Pipeline, Builder, Instructions, RAG, Fine-tuned) on OVP, finding that the Pipeline variant most consistently produces grammatical output. Both prior OVP translation systems ([Coleman et al., 2024, 2026b](#)) relied on gpt-4o and gpt-4o-mini at every LLM-driven position of the pipeline. Our contribution is to close the proprietary-dependency loop: both LLM positions (forward and backward translators) and the dataset generator that produces their training data can all be served by open-weight models, without measurable quality loss on the evaluation protocol of the previous work.

Fine-tuning open-weight LLMs for MT. Fine-tuning small open-weight LLMs for specific low-resource translation tasks has been explored with varying success ([Lankford et al., 2023](#); [Cruz and Cheng, 2019](#)), and the gains depend heavily on the availability of high-quality parallel data, which is exactly what critically endangered languages do not have. Our approach differs in that the fine-tuning data is not elicited from speakers: it is synthesized from the Yaduha language specification itself, using the backward translator to render structured targets to canonical English and a paraphrase menu to expand surface variation. The synthetic instruction-data literature ([Wang et al., 2023](#)) uses a similar bootstrapping motivation at a very different scale and for a different task. We specialize the idea to schema-constrained translation. At inference time we constrain the forward adapter’s output to the Pydantic schema with grammar-constrained decoding ([Willard and Louf, 2023](#)).

Evaluation methodology. Our evaluation methodology builds directly on [Coleman et al. \(2024, 2026b\)](#), which propose a dual-arm

¹<https://github.com/kubishi/yaduha>

back-translation protocol and a baseline contextualization scheme using unrelated-pair score distributions to establish null-hypothesis references. We adopt both and extend the evaluation with a hand-validated backward-direction reference set (Table 2) that lets the backward translator be scored directly rather than only through its effect on forward back-translation.

3 A Fully Open-Weight System

Our system consists of two LoRA-fine-tuned Qwen2.5-3B-Instruct adapters, one at each pipeline position, trained on data synthesized entirely from the language’s existing Yaduha specification. §3.1 describes the dataset-generation procedure, §3.2 the training setup, and §3.3 an end-to-end variant in which the model driving data generation is itself open-weight.

3.1 Dataset generation

Our procedure has six stages. Each stage uses only primitives exposed by the Yaduha framework, so it is language-agnostic up to the schema.

1. **Structure sampling.** We enumerate structured targets by sampling from the Sentence schema along four axes: single- vs. multi-clause (1–3 clauses per record), SV vs. SVO, plain vs. nominalized subject/object, and in-vocab vs. deliberate out-of-vocabulary content. Multi-clause sampling is important: many English inputs require multiple structured sentences to express, and models trained only on single sentences fail when asked to emit multiple.
2. **Paraphrase generation.** For each structured target, the backward translator produces an English rendering. We then query an LLM for several paraphrases of this rendering, drawn from eight transforms: *passivize*, *add_adverbial*, *add_relative_clause*, *add_irrelevant_detail*, *coordinate_and_drop*, *substitute_rare_vocab*, *idiomize*, *change_register*. The paraphrases expose the translator to a range of inputs that map to the same structured target, approximating the English-side variation it will encounter at deployment. Table 1 shows sample outputs.
3. **OOV substitution.** We curate a list of English OOV lemmas (e.g., *chihuahua*, *mansion*, *chef*) and pair each with its nearest in-vocabulary hypernym (*dog*, *house*, *cook*). Positive pairs teach the forward translator to substitute the hyper-

nym when the English input uses a rare but mappable word. Negative pairs (e.g., *laptop*, *program*) teach it to preserve the OOV lemma so the renderer emits a placeholder.

4. **Proper-noun coreference.** Proper names are always OOV and are preserved as placeholders ([Susan], [Tom]). We generate both single-mention structures and two-clause structures with explicit coreference, so that the structured target uses coreferent third-person pronouns on subsequent mentions.
5. **Backward training pairs.** For each structured target, we emit one clean and one masked backward training pair, so that a separate backward adapter can be trained to render the structured form faithfully, including preserving role-tagged placeholders verbatim. The same structured targets used for the forward adapter thus also supply the backward adapter’s training data, in a single data generation pass.
6. **Assembly.** We deduplicate on the normalized English input and produce stratified 90/10 train/-val splits that preserve the joint distribution of sentence type, pipeline stage, and transform.

3.2 Fine-tuning

We apply LoRA (Hu et al., 2022) to Qwen2.5-3B-Instruct (Qwen Team, 2024), training one adapter per direction on data synthesized by the procedure above from 750 sampled structured targets and approximately 4,500 resulting pairs, with gpt-4o-mini driving the English-generation stages. Training runs in under an hour on a single consumer GPU. The forward adapter receives English inputs and produces structured Sentences. The backward adapter receives structured Sentences and produces natural English. At inference time we decode the forward adapter with constrained generation (Willard and Louf, 2023) against the Pydantic schema to guarantee schema-valid output. Full LoRA and optimizer hyperparameters are listed in Appendix A.

3.3 End-to-end open-weight data generation

The procedure above still relies on gpt-4o-mini to render canonical English, generate paraphrases, and produce sentences for the OOV-substitution and proper-noun stages. A fully open-weight recipe requires that these calls too be served locally.

Transform	English surface
<i>Canonical: "The mules are chasing."</i>	
add_adverbial	The mules are chasing quickly.
idiomize	The mules are in hot pursuit.
substitute_rare_vocab	The mules are pursuing.
<i>Canonical: "This fish is cooking us."</i>	
passivize	We are being cooked by this fish.
add_relative_clause	This fish, which is quite large, is cooking us.
substitute_rare_vocab	This fish is preparing a meal for us.
<i>Canonical: "I write."</i>	
change_register	I am writing.
add_irrelevant_detail	I am in the process of writing.

Table 1: Sample paraphrases produced by the datagen pipeline’s paraphrase stage. Each structured target yields one canonical English rendering (via the backward translator), which is then paraphrased through the 8-transform menu. All paraphrases of a given canonical share the same structured target, so the forward translator learns to map diverse English surfaces onto one structure.

§4.4 reports the fully-open variant in which every English-generation stage is served by gpt-4o-mini and both adapters are retrained on the resulting dataset; the result matches the closed-datagen variant on COMET_c (Table 5).

4 Evaluation

Three axes of openness can vary independently in an LLM-RBMT pipeline: the **forward translator** at deployment, the **backward translator** at deployment, and the **datagen provenance** used to produce the fine-tuning corpus. We vary these one at a time. Sections 4.2–4.3 evaluate the two deployment positions with datagen provenance fixed at gpt-4o-mini; §4.4 then migrates datagen and deployment components to open-weight models progressively.

4.1 Methodology

Backward-direction protocol. We evaluate the backward translator directly before using it as a measurement instrument for the forward direction. We construct a 40-item hand-validated reference. Structured OVP sentences are programmatically sampled from the OVP vocabulary and grammar rules, stratified across simple SV, simple SVO, OOV noun, OOV verb, nominalized, and masked variants. To reduce annotation effort, we seed each English rendering with gpt-4o-mini, which performs the same structured-JSON-to-English task used at evaluation time. The authors then manually reviewed all 40 items and corrected the four rows where the seed was inaccurate². Each back-

ward model then renders the same 40 structured inputs, which we score against the gold using chrF (Popović, 2015) and reference-based COMET.

Forward-direction protocol. With the backward stage well-characterized (§4.2; gpt-4o-mini reaches 0.930 COMET against the hand-validated gold, and prior work found it back-translates OVP at ~98% accuracy at low cost (Coleman et al., 2024)), we hold it fixed at gpt-4o-mini and vary the forward translator. Given an English source s , the forward translator produces [Sentence, ...]. The backward translator then renders this structured output back to English. We compute COMET (Rei et al., 2022) (checkpoint Unbabel/wmt22-comet-da) between s and the back-translation under two treatments, as introduced in §1:

- backwards (COMET_b): the backward translator sees the full structured output, including OOV English lemmas retained in head or lemma fields.
- comparator (COMET_c): content-word lemmas are first masked to role-tagged sentinels ([NOUN], [VERB]) before the backward translator reads the structure, removing any contribution from lexical passthrough.

A successful translation scores high on both arms. We use COMET as the primary metric following Coleman et al. (2026b).

Data. The forward evaluation uses the 150-sentence set of Coleman et al. (2026b), which

sentence, the English rendering is largely determined and there is little room for stylistic variation that could systematically favor one backward model over another.

²These final gold references accurately reflect human judgment. Because the input is a simple fully-specified structured

Backward model	chrF	COMET
llama3.2:1b	0.500	0.763
llama3.2:3b	0.753	0.854
qwen2.5:3b (base)	0.791	0.869
qwen2.5:7b	0.810	0.873
llama3.1:8b	0.839	0.893
gpt-4o-mini	0.932	0.930
ft-qwen2.5:3b (ours)	0.877	0.896

Table 2: Backward-direction scores on the 40-item gold. Our fine-tuned backward adapter surpasses every off-the-shelf open-weight baseline on chrF and approaches gpt-4o-mini on both metrics. Bootstrap 95% CIs (1000 resamples, $n = 40$) span approximately ± 0.07 on chrF and ± 0.04 on COMET.

spans six sentence types (subject-verb, subject-verb-object, two-verb, two-clause, complex, and nominalization). The backward evaluation uses the 40-item hand-validated gold described above. Baseline open-weight models are llama3.2:1b, llama3.2:3b, qwen2.5:3b, qwen2.5:7b, and llama3.1:8b (Qwen Team, 2024; Llama Team, AI @ Meta, 2024), served locally with 4-bit quantization via Ollama (Ollama Project, 2025). Both evaluation sets are held out from the synthetic train/val data used to fine-tune the adapters, so there is no data leakage. All reported scores come from a single training run per configuration.

4.2 Backward-direction results

Table 2 reports scores on the 40-item hand-validated gold. gpt-4o-mini leads the off-the-shelf open-weight baselines, and the smallest model (llama3.2:1b) often substitutes fluent-but-wrong content. Applying our fine-tuning procedure to the backward direction reaches 0.877 chrF and 0.896 COMET against the gold, surpassing every off-the-shelf open-weight baseline on chrF and closing roughly 61% of the chrF gap between its Qwen2.5-3B base and gpt-4o-mini. It stays a few points behind on surface fluency, but §4.4 shows that swapping it in as the backward translator has no measurable effect on the pipeline’s forward quality.

4.3 Forward-direction results

Table 3 reports forward-direction scores for the five open-weight baselines, gpt-4o-mini, and our fine-tuned forward adapter. The fine-tuned adapter eliminates parse failures and leads on COMET_c, outperforming its base model (qwen2.5:3b) by +0.057 and gpt-4o-mini by +0.024.

Forward model	Err	COMET _b	COMET _c
llama3.2:1b	9	0.787	0.326
qwen2.5:3b (base)	15	0.771	0.415
llama3.2:3b	5	0.861	0.404
qwen2.5:7b	3	0.834	0.425
llama3.1:8b	9	0.821	0.386
gpt-4o-mini	0	0.865	0.448
ft-qwen2.5:3b (ours)	0	0.786	0.472

Table 3: Forward-direction results with gpt-4o-mini as the fixed backward evaluator. Err: number of parse failures out of 150 (the forward model failed to produce a schema-valid output on that sentence). Parse failures are excluded from all metrics rather than penalized, and no retries are attempted; the fine-tuned adapter’s zero parse failures therefore matter both for pipeline reliability and for the interpretability of its aggregate scores. Off-the-shelf open-weight baselines run at 4-bit quantization via Ollama; the fine-tuned adapter runs at bf16. Bootstrap 95% CIs (1000 resamples, $n = 150$) span approximately ± 0.035 on COMET_c; the fine-tuned adapter’s lead of +0.024 over gpt-4o-mini is consistent across resamples. Our fine-tuned adapter leads on COMET_c and is the only open-weight forward translator with zero parse failures; §4.4 shows a fully-open-datagen variant of the same recipe reaches the same COMET_c.

Table 4 gives the per-type breakdown. Our fine-tuned adapter wins or nearly ties gpt-4o-mini on all six sentence types: within ± 0.005 on simple subject-verb and nominalization, and ahead by +0.017 to +0.065 on the other four. Example 2 illustrates where these gains come from. On “*The teacher spoke and the student listened*”, gpt-4o-mini treats all four content words as OOV, whereas our fine-tuned adapter recognizes that OVP has in-vocabulary verbs for *speak* (*yadoha*) and *listen* (*naka*) and uses them, leaving only *teacher* and *student* as placeholders. The comparator arm, which masks the OOV heads, falls sharply for gpt-4o-mini (0.286) while ours holds (0.451): the gap between backwards and comparator shrinks from 0.667 to 0.433, quantifying how much less our adapter leans on lexical passthrough.

4.4 Ablations: migration to open-weight

Sections 4.3 and 4.2 fixed two of the three axes at gpt-4o-mini and varied the third. We now progressively migrate the remaining axes to open-weight models, holding the forward translator fixed at our fine-tuned Qwen2.5-3B adapter throughout and varying (a) the backward translator used at deployment and (b) the datagen provenance used

Model	SV	SVO	TV	TC	CX	NOM
qwen2.5:3b (base)	0.629	0.384	0.466	0.329	0.332	0.389
gpt-4o-mini	0.705	0.349	0.526	0.362	0.327	0.418
ft-qwen2.5:3b (ours)	0.702	0.394	0.591	0.379	0.349	0.416

Table 4: Per-sentence-type COMET_c breakdown. SV/SVO: simple single-clause; TV: two-verb; TC: two-clause; CX: complex; NOM: nominalization. Our fine-tuned forward translator wins or nearly ties gpt-4o-mini.

The teacher spoke and the student listened.	
Target	(ft-qwen2.5:3b (forward) / gpt-4o-mini (backward))
[teacher]-ii yadoha-ku. [student]-ii naka-ku.	
Backwards	
The teacher talked. The student heard.	0.884
Comparator	
The [NOUN] talked. The [NOUN] heard.	0.451

The teacher spoke and the student listened.	
Target	(gpt-4o-mini (forward) / gpt-4o-mini (backward))
[teacher]-uu [speak]-ku. [student]-uu [listen]-ku.	
Backwards	
The teacher spoke. The student listened.	0.953
Comparator	
The [NOUN] [VERB]ed. The [NOUN] [VERB]ed.	0.286

Figure 2: Two forward translations of the same input. gpt-4o-mini (bottom) passes every content word through as OOV, while our fine-tuned adapter (top) substitutes in-vocabulary OVP verbs *yadoha* (“speak”) and *naka* (“listen”).

to produce the adapter’s training data. All rows use the same 150-sentence forward evaluation set. Table 5 collects the full progression.

Row 1 is the closed-weight reference. **Row 2** swaps the forward translator to our fine-tuned adapter: COMET_c rises from 0.448 to 0.472 at zero parse failures, matching §4.3. **Row 3** additionally swaps the backward translator to our fine-tuned backward adapter, moving aggregate COMET_c by +0.001 (0.472 → 0.473) with per-type deltas within ±0.015; the two backward models produce near-identical outputs, and where they differ the fine-tuned backward is if anything *more* faithful (e.g., rendering proximity=distal as “that” rather than “the”). This null result supports **Claim 1**: the deployment pipeline runs with no proprietary-API call and no measurable loss relative to a gpt-4o-mini evaluator. **Row 4** re-trains the forward adapter on data whose every English-authoring stage is served by qwen2.5:7b rather than gpt-4o-mini; scored by the fixed gpt-4o-mini backward instrument, it reaches COMET_c 0.471, indistinguishable from the closed-

provenance version. **Row 5** then swaps in the fine-tuned backward adapter as well, so that no component touches a proprietary API. Because the evaluator is now also local, we report the 0.489 as self-evaluated quality rather than a claim against the closed-weight baseline.

5 Discussion

Under our proposed approach, the entire workflow for a new language (synthesizing training data, fine-tuning the forward and backward adapters, evaluating them, and deploying the result) runs without a proprietary API and without additional per-language linguistic labor, on consumer-GPU hardware. This is unlike prior LLM-RBMT work (Coleman et al., 2024, 2026b), which required closed-weight models at multiple pipeline positions. For communities working under CARE (Carroll et al., 2020) or OCAP principles, this makes data-sovereign language technology reachable without sacrificing translation quality. Concretely, the full recipe for OVP runs in approximately 2 GPU-hours on a single consumer card

#	Datagen provenance	Forward deploy	Backward deploy	Err	COMET _b	COMET _c
1	n/a (closed ref.)	gpt-4o-mini	gpt-4o-mini	0	0.865	0.448
2	gpt-4o-mini	ft-qwen2.5:3b	gpt-4o-mini	0	0.786	0.472
3	gpt-4o-mini	ft-qwen2.5:3b	ft-qwen2.5:3b	0	0.785	0.473
4	qwen2.5:7b	ft-qwen2.5:3b	gpt-4o-mini	0	0.760	0.471
5	qwen2.5:7b	ft-qwen2.5:3b	ft-qwen2.5:3b	0	0.778	0.489

Table 5: Progressive ablation across the three pipeline axes; the forward translator is our fine-tuned Qwen2.5-3B adapter throughout. Row 1 is the closed-weight reference; rows 2–3 deploy our open adapters with closed-weight datagen; rows 4–5 use fully-open (qwen2.5:7b) datagen, with row 5 (bold) open at every axis. All rows use the same 150-sentence forward set.

(RTX 5000 Ada): 2–3 hours of qwen2.5:7b inference for dataset generation and ~ 1 GPU-hour total for the two LoRA fine-tunes. The optional closed-weight datagen variant adds $\sim \$0.40$ of gpt-4o-mini API spend.

The backward translator’s role as a user-facing transparency layer, not only an evaluation instrument, is worth naming as a design principle. Rather than emit a fluent-but-wrong guess, the pipeline surfaces its own uncertainty: when the grammar cannot express a content word, the OVP surface shows a role-tagged placeholder and the back-translation reports it. This matters for endangered-language deployment because the community pool of speakers who could catch fluent-but-wrong output is small. The boundary between what the grammar covers and what it does not is legible from the pipeline’s own output rather than hidden inside a black-box translation. The same property lets the fine-tuned backward adapter serve as a seed for future backward-direction gold-set expansion, since a proprietary model is no longer required to draft candidates for human review.

The residual gap concentrated in the nominalization type of Table 4 is illustrative: for *The sleeper woke up suddenly*, gpt-4o-mini correctly produces a *NominalSubject* with *sleep* as the nominalized verb (*Üwi-dü-uu*, “the one who sleeps, distal”), while our fine-tuned forward translator chooses the simpler *SubjectNoun* construction with a placeholder [*sleep*] head, losing the agentive-nominalization information (the comparator arm scores 0.267 vs. 0.430). This is the clearest target for a next round of dataset iteration.

Two failure modes recur in the fully-open system’s outputs, both visible as the weakest per-type scores. First, coordinated subjects are sometimes reread as subject–object relations: “*Anna and Elsa share a bond*” collapses to a single SVO sentence [Anna]-uu [Elsa]-noka u-[share]-dü, whose

back-translation “*Anna shares Elsa*” is fluent but wrong (the comparator arm catches it, 0.271). Second, proper nouns are inconsistently preserved across coordinated clauses: “*Harry wrote and Ron read*” yields two clauses with generic third-person pronouns (Uhu mui-ku. Uhu nia-dü.). Both are clean targets for a next round of dataset augmentation.

6 Conclusion

We have presented a fully open-weight LLM-RBMT system for OVP in which the forward and backward translators are both LoRA-fine-tuned Qwen2.5-3B-Instruct adapters trained on data synthesized from the language’s existing Yahuha specification. The forward adapter matches or exceeds gpt-4o-mini on a 150-sentence forward set and the backward adapter approaches it on a hand-validated backward set, while both run locally on a single consumer GPU. We separated two senses of openness (local deployment and proprietary-free training provenance) and showed that the dataset generator itself can run on open-weight models at every English-generation stage without measurable quality loss, giving an end-to-end recipe in which no component depends on a proprietary model.

The pipeline is language-agnostic by design: both models interact exclusively with structured JSON whose field names and lemmas are English, and neither ever sees the target-language surface form. The only language-specific component is the Yahuha language package, so any Yahuha-registered language can use the same datagen and fine-tuning recipe with no model-level changes; adding a new grammatical construction to an existing language is likewise a grammar-spec change, with no changes to the training pipeline, evaluation protocol, or model architecture. Natural next steps include porting the datagen to a second Yahuha-compatible language to validate the

by-construction generalization claim empirically, scaling the training set beyond the $\sim 4,500$ pairs used here, and augmenting the datagen with the explicit conjoined-subject and proper-noun-across-clause structures identified in §5 as the fully-open system’s dominant failure modes.

7 Limitations

Single target language. Our empirical demonstration is on OVP only. The procedure is designed to port to any Yahuha-compatible language (a single per-Sentence method is the only language-specific primitive it requires), but this is supported by construction rather than empirical validation across multiple languages. Concurrent work applies the same Yahuha schema-constrained framework to five other Indigenous languages of the Americas (Bribri, Guaraní, Yucatec Maya, Orizaba Nahuatl, and Wixárika) for a cultural image-captioning shared task (Cuadros et al., 2026), placing first on human evaluation in two of them. This is evidence that the underlying schema-driven approach generalizes across typologically diverse languages, though that system pairs a vision–language model with structured decoding rather than the open-weight fine-tuning recipe evaluated here.

Inference-stack asymmetry. Open-weight baselines in Table 3 are served via Ollama with 4-bit quantization, while our fine-tuned forward translator is served in bf16 full precision during the primary evaluation. As a sanity check we also deployed the fine-tuned forward through the same Ollama pipeline as the baselines and observed a ~ 0.03 COMET_c drop, consistent with quantization effects. It still outperforms the base model served through the same stack.

No human evaluation. COMET and chrF are proxies for translation quality and faithfulness. Automatic metrics for structured-output translation in low-resource settings are an active area of research. Our characterization would be strengthened by a complementary speaker judgment study, which we did not conduct for this paper. We note that in the OVP setting, and in the extremely-low-resource setting more generally, human evaluation carries logistical constraints that are not always visible from a mainstream-MT vantage point. The evaluator pool is at most a handful of fluent speakers, so achieving meaningful inter-annotator agreement requires nontrivial evaluator training, which itself

competes for the same speakers’ time. This does not diminish the value of a speaker study, but it does mean such a study is a research project in its own right rather than a routine addition.

Scale of training and evaluation data. Our fine-tuning dataset is $\sim 4,500$ pairs synthesized from 750 sampled structures, and the evaluation sets are 150 sentences (forward) and 40 sentences (backward). These are small scales relative to standard supervised-MT benchmarks, reflecting the extremely low-resource setting.

8 Ethical Considerations

This work concerns machine translation into Owens Valley Paiute, a critically endangered Indigenous language. Tools that produce text in such a language intersect with the interests and rights of the speaker community in ways that differ from mainstream MT.

Data provenance. All target-language vocabulary and grammar rules used in this paper come from the publicly released Yahuha-OVP package, which was built in collaboration with community sources cited in Coleman et al. (2024, 2026b). We introduce no new OVP linguistic data in this paper. The 150-sentence English forward evaluation set is likewise taken unchanged from Coleman et al. (2026b), and consists of short sentences for stress-testing the translation pipeline.

Deployment and community control. A central design aim of the work is to eliminate dependencies on proprietary APIs, so that communities who wish to deploy the translator under their own data-governance policies can do so. As an ethical position, we believe that whether and how to deploy a translator for a given language should be a community decision. We present a technical method to eliminate dependencies on proprietary APIs but do not advocate for unilateral deployment of our system for any community other than through their own direction.

Quality misrepresentation. An automatic translator that produces fluent but inaccurate output can cause disproportionate harm in a language-revitalization setting, where outputs may be re-used in teaching materials or community communications and where the pool of speakers able to catch errors is small. Our pipeline is designed to make its own uncertainty visible: when the target vo-

cabulary does not cover an input word, the system produces a placeholder token rather than a fluent substitute, and the structured intermediate representation is available to any reviewer who wishes to audit what the system attempted to say.

Environmental cost. Full reproduction of the experiments reported here requires approximately two GPU-hours on a single consumer-grade accelerator and $\sim$ \$0.40 in OpenAI API calls. We consider this modest compared with both the scale of mainstream MT work and the potential benefit to the communities the method is designed to serve.

Software artifacts and licenses. All third-party software artifacts used in this work are publicly available under their respective open-source licenses: Qwen2.5-3B-Instruct (Qwen Team, 2024), the Llama 3.x baselines (Llama Team, AI @ Meta, 2024), COMET (Rei et al., 2022), sacrebleu (Post, 2018), and Ollama (Ollama Project, 2025). The Yaduha framework is released by its authors under an academic open license permitting non-commercial research and educational use with attribution. Our use of all artifacts is consistent with these licenses’ intended research use.

Release. We distinguish framework code from language-specific artifacts. The framework code, the fine-tuning scripts, and the datagen scripts will be released publicly under the Yaduha academic non-commercial license, so that practitioners working on other Yaduha-compatible languages can train their own adapters using the same recipe. The trained OVP forward and backward adapters and the OVP synthetic training datasets will be released in coordination with our partners at the Nüümü Yadoha Language Program, consistent with the community-governance principles described above.

Use of AI assistants. LLMs are a core component of the system itself and are fully documented in §3–§4. In addition, the authors used AI coding and writing assistants for coding and copy-editing. The authors accept full responsibility for all scientific claims, experiments, and analyses.

References

Stephanie Russo Carroll, Ibrahim Garba, Oscar L. Figueroa-Rodríguez, Jarita Holbrook, Raymond Lovett, Simeon Materechera, Mark Parsons, Kay Raseroka, Desi Rodriguez-Lonebear, Robyn Rowe,

Rodrigo Sara, Jennifer D. Walker, Jane Anderson, and Maui Hudson. 2020. [The CARE principles for indigenous data governance](#). *Data Science Journal*, 19(1):43.

Jared Coleman, Tainā Coleman, and Bhaskar Krishnamachari. 2026a. [RAN: Resource abundance notation for languages in NLP](#). In *Proceedings of the Sixth Workshop on NLP for Indigenous Languages of the Americas (AmericasNLP)*, pages 168–172, San Diego, California, USA. Association for Computational Linguistics.

Jared Coleman, Bhaskar Krishnamachari, Ruben Rosales, and Khalil Iskarous. 2024. [LLM-assisted rule based machine translation for low/no-resource languages](#). In *Proceedings of the 4th Workshop on Natural Language Processing for Indigenous Languages of the Americas (AmericasNLP 2024)*, pages 67–87, Mexico City, Mexico. Association for Computational Linguistics.

Jared Coleman, Ruben Rosales, Kira Toal, Diego Cuadros, Nicholas Leeds, Bhaskar Krishnamachari, and Khalil Iskarous. 2026b. [Comparing LLM-based translation approaches for extremely low-resource languages](#). In *Proceedings for the Ninth Workshop on Technologies for Machine Translation of Low Resource Languages (LoResMT 2026)*, pages 49–68, Rabat, Morocco. Association for Computational Linguistics.

Jan Christian Blaise Cruz and Charibeth Cheng. 2019. [Evaluating language model finetuning techniques for low-resource languages](#). *Preprint*, arXiv:1907.00409.

Diego Cuadros, Nicholas Leeds, Amanda Avalos, Azul Alpizar-Velazquez, Jared Coleman, Faezeh Dehghan Tarzjani, and Bhaskar Krishnamachari. 2026. [Schema-constrained image captioning for five low-resource indigenous languages](#). In *Proceedings of the Sixth Workshop on NLP for Indigenous Languages of the Americas (AmericasNLP)*, pages 257–263, San Diego, California, USA. Association for Computational Linguistics.

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 *International Conference on Learning Representations (ICLR)*.

Séamus Lankford, Haithem Afli, and Andy Way. 2023. [adaptMLLM: Fine-tuning multilingual language models on low-resource languages with integrated LLM playgrounds](#). *Information*, 14(12):638.

Llama Team, AI @ Meta. 2024. [The Llama 3 herd of models](#). *Preprint*, arXiv:2407.21783.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, et al. 2022. [No language left behind: Scaling human-centered machine translation](#). *Preprint*, arXiv:2207.04672.

- Ollama Project. 2025. *Ollama: Start building with open models*. <https://github.com/ollama/ollama>.
- OpenAI. 2024. *Introducing Structured Outputs in the API*. <https://openai.com/index/introducing-structured-outputs-in-the-api/>.
- 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.
- Matt Post. 2018. *A call for clarity in reporting BLEU scores*. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.
- Qwen Team. 2024. *Qwen2.5 technical report*. *Preprint*, arXiv:2412.15115.
- Surangika Ranathunga, En-Shiun Annie Lee, Marjana Prifti Skenduli, Ravi Shekhar, Mehreen Alam, and Rishemjit Kaur. 2023. *Neural machine translation for low-resource languages: A survey*. *ACM Computing Surveys*, 55(11):1–37.
- Ricardo Rei, José G. C. de Souza, Duarte Alves, Chrysoula Zerva, Ana C Farinha, Taisiya Glushkova, Alon Lavie, Luisa 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)*, pages 578–585, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.
- Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. *Self-instruct: Aligning language models with self-generated instructions*. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.
- Brandon T. Willard and Rémi Louf. 2023. *Efficient guided generation for large language models*. *Preprint*, arXiv:2307.09702.

A Training and Evaluation Details

LoRA adapters. We attach LoRA adapters to all linear projections of the transformer blocks of Qwen2.5-3B-Instruct (q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, down_proj; lm_head excluded). Adapter configuration: rank $r=16$, $\alpha=32$, dropout 0.05, no bias terms, CAUSAL_LM task type.

Model size comparability. Public unofficial estimates place gpt-4o-mini in the single-billion-parameter range, so our Qwen2.5-3B-Instruct base is in a comparable range (possibly a bit smaller).

Optimizer. We train with TRL’s SFTTrainer: learning rate 2×10^{-4} , cosine schedule, warmup ratio 0.1, one epoch, per-device train batch size 1 with gradient-accumulation 8 (effective batch size 8), bf16 precision, gradient checkpointing, maximum sequence length 2304, completion-only loss (loss masked to assistant tokens only), no sequence packing, and fixed seed 42. We did not perform a hyperparameter search. The values above are TRL/PEFT defaults adjusted only for memory.

Evaluation. COMET scores use the Unbabel/wmt22-comet-da checkpoint (Rei et al., 2022). chrF is computed with the default sacrebleu (Post, 2018) settings. All reported numbers are from a single training run per configuration.

Inference-time settings. All forward and backward models are run with greedy decoding (do_sample=False, temperature=0) and a maximum of 512 new tokens per call. No retries are attempted on parse failure; the failing item is excluded from all reported metrics rather than penalized. Ollama-served open-weight baselines use the same settings via the Ollama HTTP API. Proprietary gpt-4o-mini calls use temperature=0 and the OpenAI structured-outputs mode described below.

Grammar-constrained decoding. All forward models produce output that is constrained to the Pydantic sentence schema at the token level. Off-the-shelf Ollama and OpenAI models use their native structured-output APIs (Ollama’s format field with a JSON schema for open models; OpenAI’s structured-outputs mode for gpt-4o-mini (OpenAI, 2024)). The fine-tuned forward adapter uses lm-format-enforcer’s prefix_allowed_tokens_fn passed to HuggingFace generate(). In all three cases the decoder can only emit tokens compatible with the schema at each step, so parse failures reduce to cases where the model exceeds the token budget or emits a schema-legal but content-empty structure.

COMET_c validation. Because COMET was not designed for structured-output translation with placeholder sentinels, we ran a controlled experiment to check that COMET responds to content substitutions the way COMET_c needs it to. We started from 20 simple English SVO sentences (for example “The dog is eating the apple.”) and substituted 1, 2, or 3 content words under three schemes:

(a) role-tagged placeholders ([NOUN], [VERB]), (b) hypernym substitution (*dog* to *animal*, *eating* to *consuming*), and (c) random content-word substitution. COMET decreases monotonically with substitution depth under all three schemes: from a baseline of 0.973, role-tagged substitution drops to 0.563/0.384/0.275 for depths 1/2/3, hypernym substitution drops to 0.855/0.774/0.608, and random substitution drops to 0.661/0.466/0.375. Role-tagged gives the steepest drop because it removes all content information, hypernym scores are highest because semantic category is preserved, and random substitution falls in between. The same monotonic pattern holds across our 1,517 pipeline results stratified by OOV count (COMET_c 0.780 → 0.436 → 0.340 → 0.297 → 0.274 for OOV counts 0 through 4+), which confirms that COMET_c behaves as expected on the real evaluation distribution.