Hopper: A Modular Framework for Efficient Multilingual Translation Across Distant Language Families

Zachary Hamel [*1] **Darshana Upadhyay** [*2] **Eric Kitchen** [1] **Ravi Ramsaran** [1] **Srinivas Sampalli** [2]

Abstract

Large-scale multilingual models, such as Many-to-Many (M2M-100), achieve strong performance on high-resource languages; however, they remain computationally prohibitive and biased toward a small subset of language pairs. To address this, we propose Hopper, a modular neural machine translation framework that connects pre-trained, family-specific encoder–decoder pairs via a lightweight attention-based bridge, aligning latent representations without retraining the underlying models. The bridge leverages advanced attention mechanisms and gated transformation layers to enable stable adaptation between typologically diverse languages while preserving computational efficiency. We validate Hopper through two case studies, namely, French→Arabic and Russian→Arabic, demonstrating that it outperforms the 1.2B-parameter M2M-100 baseline on both language pairs despite being approximately 8.1× smaller and trained on substantially less data overall (M2M-100 uses ∼7.5B total sentence pairs across its dataset, while each bridge is trained on only 8 to 12 million). Moreover, evaluation using large language model-scoring shows higher semantic fidelity, improved syntactic alignment, and fewer hallucinations. These results demonstrate that modular bridging provides a scalable and resource-efficient approach to inclusive multilingual translation, particularly for underrepresented languages and data-scarce settings.

1. Introduction

Language translation models have become essential tools in global communication, commerce, diplomacy, and knowledge sharing. As the demand for multilingual systems increases, so does the need for translation models that are not only accurate and inclusive, but also accessible and efficient.

Most state-of-the-art translation models, such as Meta AI's open-source M2M-100 (Fan et al., 2021), aim to cover more than 100 languages in a single massive model. While effective for high-resource languages, these models face several critical limitations. With over 1.2 billion parameters, M2M-100 requires enterprise-level hardware, making it inaccessible to researchers and developers in many regions. Furthermore, M2M's training data is heavily imbalanced, with the majority of the corpus concentrated in a small number of language pairs, leaving the remaining languages with sparse and often unreliable coverage.

To address these challenges, we propose an alternative approach: building small, efficient translation models centered on individual language families, and connecting them through modular bridges. These bridges enable controlled transfer between translation families without retraining large monolithic models. This architecture, which we call *Hopper*, facilitates translation between languages (especially low-resource languages) by reusing the encoder from one family and the decoder from another, via an adapter module dubbed "the bridge". Our results show that modular bridging is a viable path toward scalable, accessible, and linguistically diverse translation systems.

1.1. Major Contributions

We summarize the key contributions of this work as follows:

- We propose *Hopper*, a modular neural machine translation framework that connects pre-trained, family-specific encoder–decoder models through a lightweight attention-based bridge, enabling cross-family translation without retraining large monolithic models. We name this architecture "Hopper" to reflect its ability to perform a single hop between distinct linguistic families' latent spaces.

[1]Nextria, Halifax, Nova Scotia, Canada [2]Faculty of Computer Science, Dalhousie University, Halifax, Nova Scotia, Canada. Correspondence to: Zachary Hamel <zachary.hamel@nextria.ca>, Darshana Upadhyay <darshana@dal.ca>.

Proceedings of the 43^{rd} International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026.

- We design an attention-driven and gated bridge architecture that aligns latent representations across typologically diverse languages while preserving computational efficiency and model stability.
- We validate the effectiveness of our framework through two challenging cross-family case studies (French→Arabic and Russian→Arabic), including translations across different scripts (Latin, Cyrillic, and Arabic).
- We show that Hopper enables accessible multilingual translation, allowing researchers and practitioners with limited computational resources to build high-quality cross-lingual systems.

1.2. Outline of the paper

The remainder of this paper is organized as follows. Section 2 reviews the related work. Section 3 presents the proposed methodology, including the Hopper architecture and the trainable bridging mechanism. Section 4 describes the experimental setup, covering training configuration, optimizer selection, activation functions, and hyperparameter tuning. Section 5 discusses the experimental results based on two case studies, namely French→Arabic and Russian→Arabic.Finally, Section 6 concludes the paper and outlines future research directions.

2. Related Work

Multilingual translation research spans three broad strategies: monolithic many-to-many models, linguistically motivated grouping, and modular adapter-based approaches. Table 1 summarizes representative work along these axes.

Monolithic multilingual models. Large shared encoder-decoder systems (Johnson et al., 2017; Aharoni et al., 2019; Fan et al., 2021) train a single model across many language pairs. While effective for high-resource languages, they require massive compute and tend to underperform on low-resource or typologically distant pairs due to data imbalance and capacity dilution.

Linguistic structure and language families. Languages form family groups that share script, morphology, syntax, and vocabulary (Comrie, 2009; Dryer & Haspelmath, 2013; Dyen et al., 1992). These shared traits simplify tokenization (Sennrich et al., 2016; Kudo & Richardson, 2018) and enable intra-family generalization (Dabre et al., 2020; Ansell et al., 2021). However, lexical overlap alone is insufficient-MSA-focused models still fail on dialectal Arabic despite shared vocabulary (Abdelali et al., 2022). Key typological differences (e.g., SVO vs. SOV word order, fusional vs. agglutinative morphology) directly affect translation quality unless architecturally accommodated.

Table 1. Summary of related approaches to multilingual translation, contrasted with Hopper.

Approach	Strategy	Key Limitation
Google Multi. (Johnson et al., 2017)	Shared enc-dec	Capacity dilution
M2M-100 (Fan et al., 2021)	Many-to-many	High compute; data imbalance
MAD-X (Pfeiffer et al., 2020)	Language adapters	Single-model only
AdapterFusion (Pfeiffer et al., 2021)	Adapter composition	No cross-family bridging
LoRA (Hu et al., 2022)	Low-rank tuning	Within-model adaptation
Hopper (ours)	**Cross-family bridge**	**Limited language pairs tested**

Modular and adapter-based methods. Rather than training monolithic models, adapter modules (Pfeiffer et al., 2020; 2021) and low-rank adaptations (Hu et al., 2022) inject small trainable parameter sets into frozen pretrained models, enabling efficient task- or language-specific specialization. These approaches reduce compute requirements but have primarily been applied within a single model rather than across independently trained encoder-decoder pairs.

Hopper builds on these foundations by combining family-specific pretrained models with a lightweight bridge that mediates representational differences between families-preserving intra-family coherence while enabling cross-family translation without retraining.

3. Methodology

3.1. Architecture

To enable cross-family translation while preserving the learned structure within each language-specific module, we designed a novel attention-based intermediary that we call the *bridge*. This component sits between frozen encoder-decoder pairs, acting as a transformation and alignment module that allows representational transfer across typologically diverse languages. Unlike monolithic multilingual systems that train end-to-end, our architecture constrains learning to a narrow, well-defined subspace-the bridge-which is solely responsible for reconciling the mismatch between the source encoder's and target decoder's language representations, as illustrated in Figure 1.

The bridge consists of a multi-layer stack of alternating self-attention and feed-forward blocks, each with architectural features tailored for linguistic alignment and efficient gradient flow. At the attention level, each block incorporates Neural Tangent Kernel (NTK)-aware rotary position

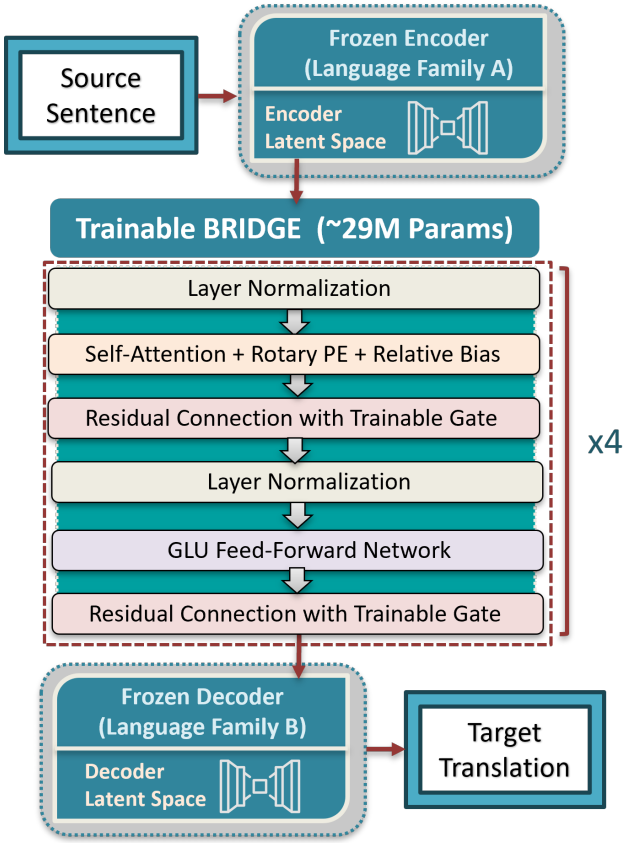

Figure 1. The Hopper architecture. A frozen encoder (Language Family A) feeds into a trainable bridge (×4 layers) with self-attention, NTK-aware rotary embeddings, GLU FFN, and gated residuals. Only the bridge's 29M parameters are trained.

embeddings (Su et al., 2021; Peng et al., 2023; Chen et al., 2023), allowing the model to retain inductive biases about sequence structure even for sequences longer than those observed during pretraining.

Formally, the rotary embedding applies a position-dependent rotation to the query and key vectors at position m:

$$\text{RoPE}(\mathbf{x}_m, m) = \mathbf{x}_m \odot \cos(m\boldsymbol{\theta}') + \text{rot}(\mathbf{x}_m) \odot \sin(m\boldsymbol{\theta}'), \tag{1}$$

where $\text{rot}(\cdot)$ interleaves and negates adjacent dimensions, and $\boldsymbol{\theta}'$ is the NTK-aware frequency vector:

$$\theta'_i = \left(b \cdot s^{d/(d-2)}\right)^{-2i/d}, \quad i = 0, 1, \ldots, \tfrac{d}{2}-1, \tag{2}$$

with base frequency b, context-length scaling factor s, and head dimension d. This formulation interpolates frequencies so that shorter-range positions retain fine resolution while longer-range positions are compressed, enabling stable extrapolation beyond fixed window sizes. This positional embedding strategy improves generalization in long-range attention contexts-a critical factor in low-resource or morphologically rich language pairs.

In addition to rotary embeddings, we incorporate a learned relative position bias mechanism, inspired by techniques used in prior sequence-to-sequence models such as T5 (Raffel et al., 2020), but extended to operate at scale across multiple attention heads (Shaw et al., 2018). This relative bias allows the attention module to remain sensitive to token-level alignment patterns even in the absence of global position anchors. Importantly, this enables the bridge to respect local ordering preferences that may differ across language families (e.g., SVO vs. SOV structures), without requiring retraining of the encoder or decoder. The full attention computation in each bridge head h is:

$$\text{Attn}^{(h)}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\tilde{\mathbf{Q}}\tilde{\mathbf{K}}^\top}{\sqrt{d_k}} + \mathbf{B}^{(h)}\right)\mathbf{V}, \tag{3}$$

where $\tilde{\mathbf{Q}} = \text{RoPE}(\mathbf{Q}, m)$ and $\tilde{\mathbf{K}} = \text{RoPE}(\mathbf{K}, m)$ are the rotary-embedded queries and keys (Equation (1)), and $B^{(h)}_{ij} = b^{(h)}_{i-j}$ is a learned scalar bias indexed by relative position for head h.

Each layer output is modulated by a trainable scalar gate applied to the residual branch. This provides a form of adaptive routing, where the model can learn to trust the output of each transformation relative to the input signal. The gating also reduces instability when training on frozen representations, ensuring that early layers of the bridge do not overly distort the semantic information passed by the encoder. The feed-forward blocks use a Gated Linear Unit (GLU) (Shazeer, 2020) with nonlinear activation and internal expansion, allowing the bridge to express complex transformations with minimal parameter overhead. Concretely, each bridge layer ℓ computes:

$$\hat{\mathbf{h}}_\ell = \mathbf{h}_{\ell-1} + \lambda^{\text{attn}}_\ell \cdot \text{Attn}(\text{LN}(\mathbf{h}_{\ell-1})), \tag{4}$$

$$\mathbf{h}_\ell = \hat{\mathbf{h}}_\ell + \lambda^{\text{ffn}}_\ell \cdot \text{FFN}(\text{LN}(\hat{\mathbf{h}}_\ell)), \tag{5}$$

where $\text{LN}(\cdot)$ denotes layer normalization and $\lambda^{\text{attn}}_\ell, \lambda^{\text{ffn}}_\ell \in \mathbb{R}$ are trainable scalar gates that modulate the residual contribution of each sub-layer. The GLU-based feed-forward network is defined as:

$$\text{FFN}(\mathbf{x}) = \big(\mathbf{W}_1\mathbf{x} \odot \sigma(\mathbf{W}_g\mathbf{x})\big)\mathbf{W}_2, \tag{6}$$

where $\mathbf{W}_1, \mathbf{W}_g \in \mathbb{R}^{d_{\text{ff}} \times d}$ and $\mathbf{W}_2 \in \mathbb{R}^{d \times d_{\text{ff}}}$ are learned projections, $\sigma(\cdot)$ is a nonlinear activation (SiLU), and $\odot$ denotes element-wise multiplication.

Overall, the bridge is lightweight, with 29M parameters-a small fraction of the parameters of the underlying models it connects. Notably, it is entirely agnostic to the training data used for the source and target models. It treats them as black-box feature extractors, learning only to align latent

spaces in a coherent manner. This architectural independence allows for flexible reuse across any number of family pair combinations, significantly expanding the scope of low-resource translation without retraining or fine-tuning full models.

The complete system, comprising a frozen encoder from one family, a frozen decoder from another, and the trainable bridge between them, is trained end-to-end with autoregressive decoding and standard cross-entropy loss (Goodfellow et al., 2016). Letting θ denote the bridge parameters, $\text{Enc}_\phi(\cdot)$ the frozen encoder, and $\text{Dec}_\psi(\cdot)$ the frozen decoder, the training objective is:

$$\mathcal{L}(\theta) = -\sum_{t=1}^{T} \log\, p_\psi\big(y_t \mid y_{<t},\ \text{Bridge}_\theta(\text{Enc}_\phi(\mathbf{x}))\big), \quad (7)$$

where $\mathbf{x}$ is the source sentence and $\mathbf{y} = (y_1, \ldots, y_T)$ is the target reference. Gradients are restricted to the bridge's 29M parameters θ (with ϕ and ψ frozen), ensuring that the pretrained linguistic priors remain intact while the bridge learns the minimal alignment required for cross-family translation. This modularity is central to Hopper's scalability and performance.

3.2. Stitching Between Families

In our proposed framework, the core idea is to enable translation between distinct language families without retraining the constituent encoder or decoder models from scratch. To achieve this, we introduce a lightweight bridging mechanism-implemented as a custom attention module-that facilitates information flow between a frozen encoder and decoder pair originating from different language families. This design ensures that the family-specific representations learned during pretraining remain intact, while the bridge adapts to imbalances in the representation space and linguistic structure between the source and target families.

This includes any Low-Rank Adaptation (LoRA) modules that have already been applied during their respective family-specific fine-tuning stages (Hu et al., 2022). Freezing these components serves two purposes: it preserves the inductive biases and linguistic specialization that each module has developed for its language family, and it significantly reduces the number of trainable parameters, making the overall system more memory and compute efficient. The only components updated during the bridging process are the parameters within the bridge module, which learns to align and transform the encoder output into a representation that the decoder can efficiently consume. The full inference pipeline can be written as:

$$\hat{\mathbf{y}} = \text{Dec}_\psi\big(\text{Bridge}_\theta(\text{Enc}_\phi(\mathbf{x}))\big), \quad (8)$$

where gradients flow only through θ during training (Equation (7)), and both ϕ (encoder) and ψ (decoder, including any previously applied LoRA weights) remain fixed.

4. Experiments

4.1. Training Configuration

The optimization strategy was carefully tuned to reflect the bridge's unique role in French-Arabic translation. We adopted a medium-to-high initial learning rate with a cosine decay schedule (Loshchilov & Hutter, 2017) and a 2,000-step warmup phase to stabilize early learning dynamics and prevent the bridge from destabilizing. Since the bridge has so few parameters, it did not warrant a lengthy warmup phase. This schedule allows the bridge to progressively learn alignment without overfitting or overwhelming the frozen representations provided by the pretrained English↔French encoder and English↔Arabic decoder.

4.2. Training Efficiency on Consumer Hardware

A key advantage of the Hopper bridge is its trainability on commodity hardware. Because only the 29M bridge parameters receive gradients while both the encoder (60M) and decoder (59M) remain frozen, GPU memory requirements are dramatically reduced compared to training a full model. Table 2 lists the hyperparameters used.

We decompose peak GPU memory into four components:

$$M_{\text{peak}} = M_{\text{weights}} + M_{\text{opt}} + M_{\text{grad}} + M_{\text{act}}. \quad (9)$$

Model weights: The frozen encoder and decoder are stored in BF16 (2 bytes per parameter), while the bridge maintains FP32 master weights (4 bytes) for numerically stable optimization:

$$M_{\text{weights}} = \underbrace{(60\text{M} + 59\text{M}) \times 2}_{\text{frozen (BF16)}} + \underbrace{29\text{M} \times 4}_{\text{bridge (FP32)}} = 0.34 \text{ GB}. \quad (10)$$

Optimizer states: AdamW maintains first- and second-moment estimates in FP32, but only for the 29M trainable bridge parameters:

$$M_{\text{opt}} = 29\text{M} \times 4 \times 2 = 0.22 \text{ GB}. \quad (11)$$

Gradients: Stored in FP32, again only for the bridge:

$$M_{\text{grad}} = 29\text{M} \times 4 = 0.11 \text{ GB}. \quad (12)$$

The static memory footprint (weights, optimizer, gradients) totals only 0.67 GB, a direct consequence of freezing the encoder and decoder and restricting trainable parameters to the bridge.

Activations: This is the dominant cost. During the backward pass, activations must be retained through both

Table 2. Bridge training hyperparameters

HYPERPARAMETER	VALUE
EPOCHS	1
BATCH SIZE	36
LEARNING RATE	5×10^{-4}
WARMUP STEPS	2,000
GRADIENT ACCUMULATION	4
BRIDGE LAYERS	4
PRECISION	BF16
MAX SEQUENCE LENGTH	256
EVAL INTERVAL	4,000 STEPS

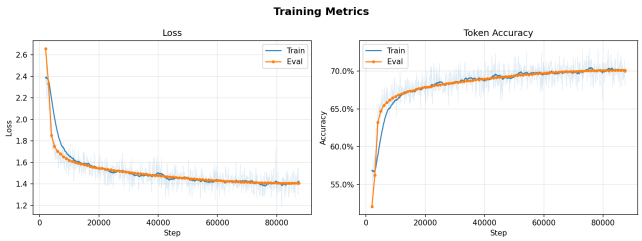

Figure 2. Training metrics for the French→Arabic bridge over a single epoch (∼87k steps). **Left:** cross-entropy loss converges from 2.7 to 1.41 (eval). **Right:** token accuracy reaches 70.1% (eval).

the bridge (4 layers, d_{ff}=4096) and the decoder (6 layers, d_{ff}=2048), since gradients flow through the frozen decoder back to the bridge. The encoder runs under `torch.no_grad()`, so its activations are not stored. Per transformer layer, the peak activation memory in BF16 for self-attention and the GLU FFN scales as $\mathcal{O}(BS(d + d_{\text{ff}} + HS))$ where B=36, S=256, d=512, and H=8. Combined with the output logits ($B \times S \times V \times 4$ bytes in FP32, where V=50,000):

$$M_{\text{act}} \approx \underbrace{4 \times 0.47}_{\text{bridge}} + \underbrace{6 \times 0.58}_{\text{decoder}} + \underbrace{1.84}_{\text{logits}} \approx 7.20 \text{ GB}. \quad (13)$$

Total theoretical peak:

$$M_{\text{peak}} \approx 0.34 + 0.22 + 0.11 + 7.20 = 7.87 \text{ GB}. \quad (14)$$

In practice, CUDA context initialization, PyTorch's caching memory allocator, and `torch.compile` kernel buffers add overhead, but the empirical peak remains well below 24 GB, fitting on a single NVIDIA RTX 3090. By contrast, training all 148M parameters end-to-end would require ∼5× more optimizer and gradient memory alone, with corresponding increases in activation storage.

The effective batch size of 144 (micro-batch 36 × gradient accumulation 4) further improves memory efficiency: only 36 sequences reside in GPU memory at any given time, while the larger effective batch stabilizes gradient estimates.

Figure 2 presents the full training dynamics over approximately 87,000 steps. The loss curve (left) shows convergence from an initial cross-entropy of 2.7 to a final train loss of 1.55 and eval loss of 1.41, with evaluation loss closely tracking training loss throughout, indicating no overfitting. Token-level accuracy (right) rises from 55% to a final train accuracy of 68.1% and eval accuracy of 70.1%, with the two curves closely aligned throughout training.

These results confirm that the bridge can be trained end-to-end in a single pass over 12M sentence pairs on a single 24 GB consumer GPU, making the approach accessible to independent researchers and institutions without enterprise-level compute infrastructure.

4.3. Data

For training data, we curated high-quality parallel corpora from No Language Left Behind (NLLB) (Costa-jussà et al., 2022) and the United Nations Parallel Corpus (UNPC) (Ziemski et al., 2016). The French→Arabic bridge was trained on approximately 12 million sentence pairs, while the Russian→Arabic bridge used 8 million sentence pairs with identical filtering procedures. Tokenization uses Byte-Pair Encoding (BPE) (Sennrich et al., 2016) in accordance with the underlying source and target models. Both corpora cover a wide variety of domains to encourage robustness across topics and registers, ensuring that bridges can generalize to unseen contexts. All models were trained using standard cross-entropy loss (Goodfellow et al., 2016), computed over the decoder's output tokens against the Arabic reference translations. Despite its simplicity, cross-entropy remains a strong baseline for supervised translation tasks and allowed us to focus on the behavior and performance of the bridging mechanism itself.

This approach of coupling frozen encoder-decoder pairs through a learned bridge offers several advantages: it enables reuse of high-quality pretrained models without retraining them end-to-end, facilitates cross-family translation with minimal data, and allows extension to new language pairs by simply training a new bridge.

5. Results and Discussion

In this section, we present and analyze the experimental results for two case studies, namely, French→Arabic and Russian→Arabic. Each case study is examined separately to highlight the performance and insights specific to the corresponding language pair. A detailed description and discussion of the results for each case study are provided in the following subsections.

Table 3. LLM-judge scores (%, higher is better) on 100 samples of FLORES-101 devtest. Hopper outperforms M2M-100 (1.2B) across all dimensions despite being 8.1× smaller.

METRIC (%)	HOPPER	M2M-100
ACCURACY	85.4	49.1
FLUENCY	86.4	59.7
TONE/STYLE	86.5	60.2
OVERALL	86.1	56.3

5.1. Case Study 1: French→Arabic Translation

Despite using only 148M total inference parameters (119M encoder-decoder + 29M bridge), Hopper is approximately 8.1× smaller than M2M-100 (1.2B) and trains only the 29M bridge parameters (approximately 41× fewer trainable parameters than full fine-tuning of a 1.2B model). Our bridge-based architecture demonstrated superior translation quality on French-Arabic tasks when evaluated using ChatGPT-4o. Both models were decoded using beam search with a beam size of 5, matching the configuration used in the M2M paper.

Evaluation was conducted across three dimensions: accuracy, fluency, and tone/style (professional register, consistent with the journalistic nature of the FLORES-101 source texts), each scored on a percentage scale from 0% (poor) to 100% (human-like). We report results on the first 100 (non-randomized) sentences of the FLORES-101 devtest set (Goyal et al., 2022). As shown in Table 3, Hopper achieved higher scores than M2M-100 across all categories: 85.4% vs. 49.1% for accuracy, 86.4% vs. 59.7% for fluency, and 86.5% vs. 60.2% for tone/style, with an overall score of 86.1% compared to 56.3%. These gains demonstrate that Hopper consistently delivers more faithful, fluent, and natural translations compared to the much larger M2M-100 model.

Table 4 presents selected qualitative examples from the FLORES-101 devtest set where Hopper produces faithful translations while M2M-100 introduces severe errors including lexical hallucinations, semantic distortions, and mistranslation of key terms.

In addition to Large Language Model (LLM)-based human-aligned evaluations, we also report automatic evaluation using the Metric for Evaluation of Translation with Explicit ORdering (METEOR) (Banerjee & Lavie, 2005) on the same 100-sentence subset. We select METEOR over BLEU as our automatic metric because METEOR incorporates synonym matching, stemming, and explicit word-order modeling, yielding significantly higher correlation with human judgments (Lavie & Agarwal, 2007); empirical evaluations at WMT shared tasks have consistently shown METEOR outperforms BLEU in system-level correlation (Callison-Burch et al., 2006; Reiter, 2018). Hopper achieved a METEOR score of 0.3852, significantly higher than the 0.2564 achieved by M2M-1.2B (+0.1288 delta). Notably, the LLM-based evaluation shows a larger performance gap (86.1% vs. 56.3%) than METEOR (+0.129 delta). This discrepancy arises because METEOR awards partial credit for lexical overlap even when translations contain semantic errors or hallucinations, whereas LLM judges penalize meaning-level failures more severely, as evidenced by the qualitative examples in Table 4, where M2M-100 produces superficially plausible but semantically incorrect outputs. The consistency in directionality between both metrics provides additional evidence that the modular bridge improves cross-family translation quality while maintaining efficiency.

5.2. Case Study 2: Russian→Arabic Translation

To demonstrate that the Hopper bridging approach generalizes beyond a single language pair, we conducted a second case study on Russian→Arabic translation. This pairing presents distinct challenges: Russian uses Cyrillic script, has a different morphological structure (fusional with extensive case marking), and belongs to the Slavic language family, typologically distant from both French (Romance) and Arabic (Semitic).

We trained a new bridge connecting a frozen English↔Russian encoder with the same frozen English↔Arabic decoder used in the French experiment, using 8 million sentence pairs from NLLB with the same filtering procedure applied to the French→Arabic corpus. Notably, the Russian encoder uses Grouped Query Attention (GQA) (Ainslie et al., 2023), while the French encoder and Arabic decoder use standard Multi-Head Attention (MHA). This architectural heterogeneity poses no obstacle: the bridge learns to transform encoder outputs into the representational format expected by the decoder, effectively decoupling the internal attention mechanisms of the source and target models. This demonstrates that Hopper can connect models with fundamentally different architectures, enabling flexible composition of independently trained components. Figure 3 shows the training dynamics, which exhibit similar convergence behavior to the French→Arabic bridge. On the same 100-sentence FLORES-101 evaluation subset, the Russian→Arabic bridge achieved a METEOR score of 0.3171 compared to 0.2382 for M2M-100 (+0.0789 delta).

We also conducted LLM-based evaluation using the same protocol as the French→Arabic experiment. As shown in Table 5, Hopper achieved higher scores than M2M-100 across all categories: 83.1% vs. 46.9% for accuracy, 83.6% vs. 62.0% for fluency, and 83.7% vs. 62.6% for tone/style, with an overall score of 83.5% compared to 57.2%. These results mirror the French→Arabic findings, confirming that Hopper delivers more faithful, fluent, and natural translations across

Table 4. Translation examples (French→Arabic) as English back-translations. M2M-100 hallucinations in bold.

Source (glossed)	Hopper	M2M-100
We now have 4-month-old **mice** that are not diabetic whereas they were before.	We now have four-month-old **mice** that are not diabetic as they were before.	We now have 4-month-old **employees** who don't have diabetes.
The film, starring Ryan Gosling and **Emma Stone**, received nominations in all major categories.	The film with Ryan Gosling and **Emma Stone** received nominations in all major categories.	The film with Ryan Gosling and **Amazon Stone** won an award in all major categories.
Siminoff said sales increased after appearing on **Shark Tank** where the panel refused to fund the **start-up**.	Siminoff said sales increased after appearing on **Shark Tank** where the panel refused to fund the **start-up**.	Siminoff said sales increased after appearing on the **wheat truck** where the jury refused funding.

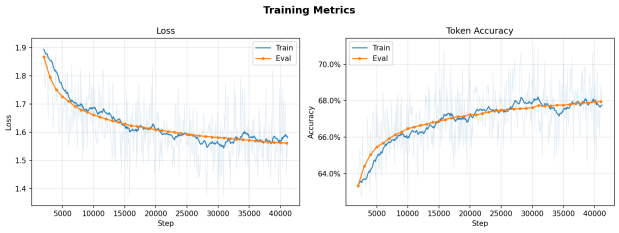

Figure 3. Training metrics for the Russian→Arabic bridge. The loss and accuracy curves show stable convergence similar to the French→Arabic experiment.

Table 5. LLM-judge scores (%, higher is better) on 100 samples of FLORES-101 devtest for Russian→Arabic. Hopper again outperforms M2M-100 across all dimensions.

Metric (%)	Hopper	M2M-100
Accuracy	83.1	46.9
Fluency	83.6	62.0
Tone/Style	83.7	62.6
Overall	83.5	57.2

different source language families. Table 6 presents selected qualitative examples from the Russian→Arabic evaluation where Hopper produces faithful translations while M2M-100 introduces severe errors including lexical hallucinations and mistranslation of key terms. Table 7 summarizes METEOR results across both case studies.

These results confirm that the modular bridging approach transfers effectively to new source languages without architectural modification; only the encoder and bridge weights differ between experiments, while the decoder remains fixed.

5.3. LLM as Judge

We supplement reference-based automatic metrics with LLM-based evaluation because recent shared tasks show that learned neural Machine Translation (MT) metrics correlate much better with human judgments than n-gram overlap metrics such as Bilingual Evaluation Understudy (BLEU) (Cífka & Bojar, 2018) or Recall-Oriented Understudy for Gisting Evaluation (ROUGE); concurrently, LLMs like GPT-4 have become competitive evaluators that match or exceed top neural metrics against human Multidimensional Quality Metrics (MQM) labels (Freitag et al., 2022; 2023; Kocmi & Federmann, 2023).

Classic automatic metrics such as BLEU and ROUGE measure translation quality primarily through n-gram overlap with reference outputs. While effective for assessing lexical similarity, these methods have well-documented limitations: they are insensitive to meaning preservation when different but semantically equivalent lexical choices are used, they do not reliably capture grammatical accuracy, and they entirely omit stylistic and tonal considerations. These gaps are particularly pronounced in low-resource, morphologically rich, or syntactically divergent language pairs, where reference translations may exhibit high variability.

In contrast, large language models such as ChatGPT-4o are capable of evaluating translations in a manner that aligns more closely with human judgment. Specifically, LLM-based ranking can: (1) assess semantic fidelity beyond surface form, ensuring nuanced meaning is preserved even under paraphrasing; (2) evaluate syntactic coherence and morphological agreement directly in the target language; (3) judge pragmatic and stylistic alignment, including tone and style; and (4) provide consistent evaluation across both in-domain and out-of-domain test sets, where overlap-based metrics often fail to correlate with perceived quality.

This ability to jointly consider semantic, syntactic, and pragmatic dimensions makes LLM-based evaluation a more comprehensive and representative benchmark for real-world translation performance. As such, it serves as a robust complement-or, in certain contexts, alternative-to traditional automatic metrics, especially when high-fidelity qualitative evaluation is needed at scale.

Access method and environment: We used the ChatGPT-4o API (not the Chat interface), setting a strict system prompt and temperature of 0.0 to improve reproducibility. This environment is managed and optimized by OpenAI (e.g., platform-level safety and performance interventions,

Table 6. Translation examples (Russian→Arabic) as English back-translations. M2M-100 hallucinations in bold.

Source (glossed)	Hopper	M2M-100
In the **1960s**, Brzezinski served as an advisor to John F. Kennedy, then in the Johnson administration.	In the **1960s**, Brzezinski served as an advisor to John Kennedy, then in the Johnson administration.	In the **1980s**, Brzezinski was an official under John Fing Kennedy, then in the Landon Birzon administration.
The **ZMapp antibody cocktail** initially showed promising results in this area.	The **ZMapp antibody cocktail** initially showed promising results in this area.	The **cocaine from ZMapp antibodies** initially showed positive results in this area.
He was **robbed by pirates**, attacked by a rabid dog in Tibet, and was arrested in India.	He was **robbed by pirates**, attacked by a rabid dog in Tibet, and was arrested in India.	He was **assassinated by Brazilians**, beaten by an angry dog in Tibet, and was arrested in India.

Table 7. METEOR scores across both case studies. Hopper consistently outperforms M2M-100 on cross-family translation tasks.

LANGUAGE PAIR	HOPPER	M2M-100	DELTA
FRENCH→ARABIC	0.385	0.256	+0.129
RUSSIAN→ARABIC	0.317	0.238	+0.079

model hosting, and potential context enhancements over time), so we treat it as a black box with fixed external controls. We logged the model label ("GPT-4o") in the appendix.

Protocol: For each source sentence, the judge received two anonymized translations ("system A/B") and returned integer ratings in [1,10] for accuracy, fluency, and tone/style. System identities were hidden and A/B randomized to mitigate documented biases for LLM-as-judge. The full judge prompt is provided in Appendix A.

The LLM-based evaluation indicated that the bridge consistently produced translations with: (1) improved semantic fidelity, preserving nuanced meaning across distant language families; (2) better syntactic alignment, effectively handling long-distance dependencies and morphological agreement; and (3) reduced hallucination rate, with fewer instances of introducing extraneous or culturally mismatched content. Performance gains over M2M-1.2B were more pronounced on out-of-domain and low-resource topics, suggesting that the bridge benefits from the domain robustness of its underlying monolingual family models while learning efficient cross-family mappings.

6. Conclusion

We have presented Hopper, a modular neural machine translation framework that enables efficient translation across distant language families through a lightweight attention-based bridge. By connecting frozen, family-specific encoder-decoder pairs without retraining the underlying models, Hopper achieves superior translation quality compared to the much larger M2M-100 baseline while using approximately 8.1× fewer parameters (148M vs. 1.2B).

Our two case studies demonstrate that the bridge successfully mediates representational differences across typologically diverse languages and scripts (Latin, Cyrillic, Arabic). On French→Arabic, Hopper achieved an overall LLM-judge score of 86.1% compared to 56.3% for M2M-100 (+29.8 percentage points) and a METEOR score of 0.385 vs. 0.256 (+0.129). On Russian→Arabic, Hopper scored 83.5% overall vs. 57.2% (+26.3 percentage points) with a METEOR score of 0.317 vs. 0.238 (+0.079). Across both pairs, Hopper outperformed M2M-100 on every evaluation dimension-accuracy, fluency, and tone/style-confirming that modular bridging produces translations with higher semantic fidelity, better syntactic alignment, and fewer hallucinations than monolithic multilingual models.

Future Work: Two extensions are planned: (1) injecting targeted LoRA modules (Hu et al., 2022) into the frozen encoder and decoder to allow lightweight adaptation for structurally divergent pairs (Pfeiffer et al., 2020), and (2) introducing multiple pivots or chained bridges to reduce semantic drift from single-pivot reliance (Dabre et al., 2020).

Impact Statement

This paper presents work whose goal is to advance machine translation through a modular and computationally efficient architecture. By reducing computational requirements by approximately $8.1\times$ compared to large monolithic models, our approach improves accessibility to high-quality translation systems, particularly for researchers and practitioners with limited resources. This supports broader participation in multilingual NLP, enables deployment in low-resource or privacy-sensitive environments, and reduces energy consumption and environmental impact.

Our work can benefit applications in education, healthcare, public services, and humanitarian contexts by enabling on-device and air-gapped translation where cloud-based solutions are infeasible, while improved support for underrepresented language pairs helps reduce linguistic barriers and promote inclusivity.

The risks of our work are typical for machine translation:

Hopper may produce mistranslations or hallucinations, so we emphasize responsible deployment with domain-specific evaluation and human oversight in high-stakes applications.

References

Abdelali, A., Durrani, N., Dalvi, F., and Sajjad, H. Post-hoc analysis of Arabic transformer models. In *Proceedings of the Fifth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pp. 259–272, 2022.

Aharoni, R., Johnson, M., and Firat, O. Massively multilingual neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3874–3884, 2019.

Ainslie, J., Lee-Thorp, J., de Jong, M., Zemlyanskiy, Y., Lebrón, F., and Sanghai, S. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 4895–4901, 2023.

Ansell, A., Ponti, E. M., Pfeiffer, J., Ruder, S., Glavaš, G., Vulić, I., and Korhonen, A. MAD-G: Multilingual adapter generation for efficient cross-lingual transfer. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 4762–4781, 2021.

Banerjee, S. and Lavie, A. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pp. 65–72, 2005.

Callison-Burch, C., Osborne, M., and Koehn, P. Re-evaluating the role of BLEU in machine translation research. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 249–256, 2006.

Chen, S., Wong, S., Chen, L., and Tian, Y. Extending context window of large language models via positional interpolation. *arXiv preprint arXiv:2306.15595*, 2023.

Cífka, O. and Bojar, O. Are BLEU and meaning representation in opposition? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1362–1371, 2018.

Comrie, B. *The World's Major Languages*. Routledge, 2nd edition, 2009.

Costa-jussà, M. R., Cross, J., Çelebi, O., Elbayad, M., Heafield, K., Heffernan, K., Kalbassi, E., Lam, J., Licht, D., Maillard, J., et al. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*, 2022.

Dabre, R., Chu, C., and Kunchukuttan, A. A survey of multilingual neural machine translation. *ACM Computing Surveys*, 53(5):1–38, 2020.

Dryer, M. S. and Haspelmath, M. WALS online (v2020.4). Zenodo, 2013. `https://wals.info`.

Dyen, I., Kruskal, J. B., and Black, P. An Indoeuropean classification: A lexicostatistical experiment. *Transactions of the American Philosophical Society*, 82(5):1–132, 1992.

Fan, A., Bhosale, S., Schwenk, H., Ma, Z., El-Kishky, A., Goyal, S., Baines, M., Celebi, O., Wenzek, G., Chaudhary, V., et al. Beyond English-centric multilingual machine translation. *Journal of Machine Learning Research*, 22 (107):1–48, 2021.

Freitag, M., Rei, R., Mathur, N., Lo, C.-k., Stewart, C., Avramidis, E., Kocmi, T., Foster, G., Lavie, A., and Martins, A. F. Results of the WMT22 metrics shared task: Stop using BLEU—neural metrics are better and more robust. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pp. 46–68, 2022.

Freitag, M., Mathur, N., Lo, C.-k., Avramidis, E., Rei, R., Thompson, B., Kocmi, T., Blain, F., Deutsch, D., Stewart, C., et al. Results of the WMT23 metrics shared task: Metrics might be guilty but references are not innocent. In *Proceedings of the Eighth Conference on Machine Translation*, pp. 578–628, 2023.

Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning*. MIT Press, 2016.

Goyal, N., Gao, C., Chaudhary, V., Chen, P.-J., Wenzek, G., Ju, D., Krishnan, S., Ranzato, M., Guzmán, F., and Fan, A. The FLORES-101 evaluation benchmark for low-resource and multilingual machine translation. *Transactions of the Association for Computational Linguistics*, 10:522–538, 2022.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Johnson, M., Schuster, M., Le, Q. V., Krikun, M., Wu, Y., Chen, Z., Thorat, N., Viégas, F., Wattenberg, M., Corrado, G., et al. Google's multilingual neural machine translation system: Enabling zero-shot translation. In *Transactions of the Association for Computational Linguistics*, volume 5, pp. 339–351, 2017.

Kocmi, T. and Federmann, C. Large language models are state-of-the-art evaluators of translation quality. In *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*, pp. 193–203, 2023.

Kudo, T. and Richardson, J. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 66–71, 2018.

Lavie, A. and Agarwal, A. METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pp. 228–231, 2007.

Loshchilov, I. and Hutter, F. SGDR: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2017.

Peng, B., Quesnelle, J., Fan, H., and Shao, E. YaRN: Efficient context window extension of large language models. *arXiv preprint arXiv:2309.00071*, 2023.

Pfeiffer, J., Vulić, I., Gurevych, I., and Ruder, S. MAD-X: An adapter-based framework for multi-task cross-lingual transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pp. 7654–7673, 2020.

Pfeiffer, J., Kamath, A., Rücklé, A., Cho, K., and Gurevych, I. AdapterFusion: Non-destructive task composition for transfer learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 487–503, 2021.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21 (140):1–67, 2020.

Reiter, E. A structured review of the validity of BLEU. *Computational Linguistics*, 44(3):393–401, 2018.

Sennrich, R., Haddow, B., and Birch, A. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1715–1725, 2016.

Shaw, P., Uszkoreit, J., and Vaswani, A. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 464–468, 2018.

Shazeer, N. GLU variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

Su, J., Lu, Y., Pan, S., Murtadha, A., Wen, B., and Liu, Y. RoFormer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*, 2021.

Ziemski, M., Junczys-Dowmunt, M., and Pouliquen, B. The United Nations parallel corpus v1.0. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pp. 3530–3534, 2016.

A. LLM Judge Prompt

The following prompt was used for LLM-based evaluation with ChatGPT-4o:

```
SYSTEM_PROMPT = "You are a professional
↪  bilingual translation evaluator (French
↪  -> Arabic).
You judge translations objectively and
↪  consistently.
You do not invent context.
You strictly follow the scoring rubric and
↪  output valid JSON only."

USER_PROMPT_TEMPLATE = "
You will evaluate two Arabic translations
↪  of a French source text.

SOURCE (French):
{source}

TRANSLATION A (Arabic):
{translation_a}

TRANSLATION B (Arabic):
{translation_b}

Evaluate each translation on Accuracy,
↪  Fluency, and Tone & Style (0-10 each).

Return JSON in exactly this format:

{
  \"source_language\": \"fr\",
  \"target_language\": \"ar\",
  \"scores\": {
    \"A\": {
      \"accuracy\": 0,
      \"fluency\": 0,
      \"tone_style\": 0,
      \"overall\": 0
    },
    \"B\": {
      \"accuracy\": 0,
      \"fluency\": 0,
      \"tone_style\": 0,
      \"overall\": 0
    }
  },
```

```
  \"overall_method\": \"overall =
  ↪  round((accuracy + fluency +
  ↪  tone_style) / 3, 2)\",
  \"winner\": \"A_or_B_or_Tie\",
  \"rationale\": {
    \"key_differences\": [\"...\"],
    \"summary\": \"Brief explanation\"
  },
  \"confidence\": 0.0
}
"
```