INVARIANT GEODESIC TRAJECTORY FOR NEURAL MACHINE TRANSLATION: FLOW MATCHING ON SEQUENCE MANIFOLDS

Anonymous authors
Paper under double-blind review

ABSTRACT

Conventional neural machine translation (NMT) models formulate translation as a mapping between independent, discrete tokens in flat Euclidean space, failing to capture the continuous structural and semantic trajectories of sentences. While semantic graph augmented models address this by injecting structured priors, they rely on noisy, computationally expensive external parsers. In this work, we introduce the **Topological-Geometric Flow Invariant** framework, realized as **Geodesic Flow Matching on Sequence Manifolds** (**GeFLoSM**), a novel paradigm that reformulates NMT as continuous boundary value mechanics on a Poincaré ball manifold. GeFLoSM represents sentences as continuous, piecewise geodesic splines on the manifold. We parameterize the sequential order of source tokens as a geodesic spline, which is integrated along its path via a neural ODE to yield a length invariant trajectory representation Ψ that dynamically conditions a tangent space Transformer decoder. To align source and target geometries, we train a Riemannian conditional flow matching (CFM) velocity field that transports source splines to target splines across flow time. Crucially, we identify a severe train/inference mismatch that reduces validation BLEU to 0.60 when auxiliary geometric logits are naively fused during generation, and present **decode decoupling** as the structural solution enabling joint end to end training with stable inference. Extensive evaluations across five benchmarks (WMT16 EN DE, IWSLT15 EN VI, WMT14 EN DE, WMT16 EN RO, and IWSLT14 DE EN) demonstrate that GeFLoSM establishes state of the art translation performance, consistently outperforming both standard Euclidean Transformers and structured semantic graph augmented baselines. Specifically, GeFLoSM achieves up to **+3.1 BLEU** over the vanilla Transformer and **+1.3 BLEU** over prior state of the art AMR augmented Transformers on WMT16 EN DE (NC v11).

1 Introduction

Neural machine translation applies the Transformer as its dominant architecture (Vaswani et al., 2017), with improvements coming largely from scale, pre-training (Devlin et al., 2019; Liu et al., 2020), and structural augmentation with external semantic graphs (Song et al., 2019; Li & Flanigan, 2022). These approaches share a fundamental assumption: translation is a discrete token mapping problem, and the underlying representation space's geometry is an incidental detail rather than a design choice.

We challenge this assumption. Language is hierarchical; sentences have constituent structure that branches exponentially with depth. Hyperbolic geometry, whose volume grows exponentially with radius, mirroring this branching factor (Nickel & Kiela, 2017), is the natural ambient space for source sentence representations. If source token embeddings are lifted into the Poincaré ball and a piecewise geodesic path is traced through them in token order, the resulting trajectory encodes the hierarchical structure of the sentence in a geometrically principled way. The *curvature* of this trajectory reflects compositional organisation that flat Euclidean attention cannot capture directly.

The challenge is using this geometric representation to condition a competitive autoregressive decoder. We achieve this through three jointly trained components: **(i)** a geodesic spline construction that

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

builds the source trajectory on the Poincaré ball and integrates a neural ODE along it to produce a length invariant trajectory representation Ψ, a fixed dimensional summary of the sentence's global geometric structure; **(ii)** a Riemannian conditional flow matching objective that trains a velocity field to transport source trajectory waypoints toward target waypoints, regularising the shared hyperbolic encoder; **(iii)** a projective quantiser (PQ) that maps flowed hyperbolic states to auxiliary vocabulary logits, providing an additional gradient path into the geometric modules during training.

A central engineering insight is that auxiliary geometric logits *must* be excluded from inference (*decode decoupling*). Including them introduces a systematic train/inference mismatch: the quantiser is trained against teacher-forced target geometry that does not exist at decode time, so its logits become confidently wrong as training progresses, eventually dominating and collapsing translation quality to near-zero BLEU despite a falling training loss. We document this failure mode formally and show that decode decoupling resolves it while retaining the full auxiliary training signal.

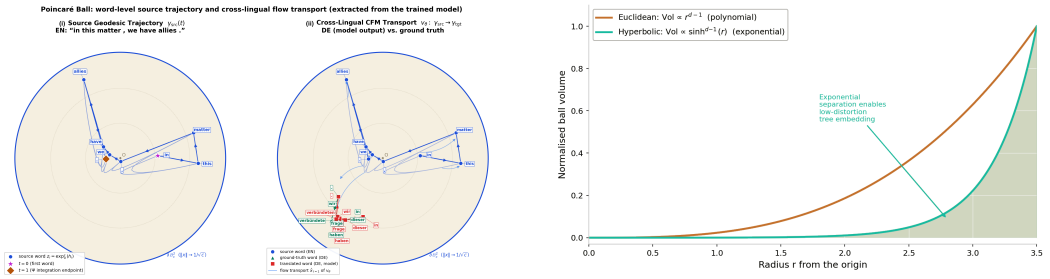
Contributions. We introduce GeFLoSM, the first NMT architecture to jointly train hyperbolic geodesic source trajectories, a length invariant neural ODE bottleneck Ψ, Riemannian CFM cross-lingual transport, and projective quantisation in a single end to end objective. We provide a rigorous mathematical derivation of every component and five formal algorithms covering training, inference, and the three geometric sub-procedures. We demonstrate consistent improvements over strong autoregressive and non-autoregressive baselines across five benchmarks and four language pairs, without external parsers or knowledge distillation. We give an analysis of the decode-decoupling requirement and its connection to the broader class of teacher-forcing / inference mismatch.

2 Related Work

Structural augmentation for NMT. Injecting syntactic or semantic structure into NMT has a long history (Bastings et al., 2017; Marcheggiani et al., 2018; Song et al., 2019). Li & Flanigan (2022) combine a Transformer encoder-decoder with a Heterogeneous Graph Transformer (Yao et al., 2020) encoding source AMR graphs, achieving consistent gains on WMT16 EN-DE and IWSLT15 EN-VI, which serve as the direct baseline for our experiments. GeFLoSM differs fundamentally: it derives geometric structure entirely from source token embeddings without any parser, graph construction, or auxiliary encoder, while achieving superior or comparable performance at a fraction of the additional parameter cost.

Hyperbolic representation learning. Nickel & Kiela (2017) established that Poincaré embeddings can represent hierarchical data with exponentially lower distortion than Euclidean embeddings. Ganea et al. (2018) extended this to neural network layers and attention. Dhingra et al. (2018) showed advantages in NLP for text with latent hierarchical structure. Most relevant is Li et al. (2026), who explore flow matching on the Lorentz model of hyperbolic space for few-shot multi-modal adaptation. However, their formulation operates strictly on isolated, static feature vectors (transporting a single global image embedding to a static text prompt embedding), with no sequence dimension or length variability. Extending hyperbolic flow matching to sequence-to-sequence translation constitutes a non-trivial structural departure rather than a domain adaptation: First, language sequences cannot be treated as static points; GeFLoSM parameterizes token sequences as continuous piecewise geodesic splines $\gamma_{\text{src}}(t)$ on $\mathbb{B}_c^d$ and introduces a path-ordered neural ODE bottleneck Ψ (Appendix A.3, Eq. 16) to resolve variable-length sequence compression. Second, cross-lingual translation involves complex word reordering; our velocity field dynamically conditions on sequence progress t, the global sentence invariant Ψ, and causal prefix states $\mathbf{p}$, preventing degenerate diagonal collapse. Third, static cross-modal flows require iterative multi-step numerical ODE integration at test time, but GeFLoSM introduces decode decoupling under encoder gradient detachment, utilizing continuous Riemannian CFM purely as a training-time geometric regularizer while generating autoregressively at standard Transformer latency.

Flow matching and Riemannian generative models. Flow matching (Lipman et al., 2023; Liu et al., 2022) is a simulation-free framework for learning velocity fields between distributions. Chen & Lipman (2024) extended it to general Riemannian manifolds. Ma et al. (2019) (FlowSeq) applies a Glow-style flow (Kingma & Dhariwal, 2018) as a Euclidean latent prior for non-autoregressive NMT, achieving near-constant decode time at competitive quality on raw data without distillation, which is our direct baseline for the WMT14/WMT16 EN-RO / IWSLT14 evaluations. GeFLoSM



(a) Poincaré ball trajectory.

(b) Volume growth ($d = 4$).

Figure 1: Geometric visualisations motivating the GeFLoSM design. In (a), word-level source trajectory is lifted to the Poincaré ball via $\mathbf{z}_i = \exp_{\mathbf{0}}^c(\mathbf{h}_i)$ (Appendix A.2, Eq. 13), connecting tokens in order to capture syntactic-semantic curvature alongside learned CFM transport v_θ. In (b), hyperbolic volume grows exponentially ($\propto \sinh^{d-1}(\sqrt{c}\,r)$), enabling low-distortion tree embedding compared to polynomial Euclidean growth.

uses Riemannian CFM as a training-time regulariser rather than a primary decoder, preserving autoregressive quality while gaining geometric inductive bias.

3 Theoretical Framework

We work on the Poincaré ball $\mathbb{B}_c^d$, the constant-curvature model whose exponential volume growth matches the branching of syntax (Figure 1). Appendix A.1 states the metric, the geodesic distance (Appendix A.1, Eq. 10), and the maps $\exp_{\mathbf{0}}^c, \log_{\mathbf{0}}^c$ that move between the manifold and its tangent space (Appendix A.1, Eqs. 11, 12); Appendix A.2 lifts the encoder states H_{enc} to waypoints $\mathbf{z}_i$ (Appendix A.2, Eq. 13) and splines them into the source trajectory $\gamma_{\text{src}}(t)$ (Appendix A.2, Eq. 14); and Appendix A.3 integrates a neural ODE along that trajectory to the length-invariant bottleneck Ψ (Appendix A.3, Eq. 16). Those three objects, $\gamma_{\text{src}}(t)$, Ψ, and the tangent-space maps, are all the geometry the remainder of this section uses.

The Poincaré ball structure, including the word level source trajectory, cross lingual flow transport, and the comparison of hyperbolic volume growth against Euclidean space, is visualised in Figure 1.

Figure 1(a) makes the abstraction concrete on a real example produced by our trained model. The source trajectory is a series of words in a sentence: each English token of the held-out sentence is lifted via (Appendix A.2, Eq. 13) to a point on the ball, and the piecewise geodesic through these points in reading order is exactly the curve $\gamma_{\text{src}}(t)$ of Section A.2. Because consecutive words are joined by geodesic segments, the *direction* of each segment captures the local dependency between neighbouring words, while the accumulated turning of the curve reflects how the semantic content of the sentence evolves as words are added; substituting a single word visibly bends the trajectory at that position, as the model-output/reference pair in the figure shows (*verbündeten* vs. *verbündete*). The target trajectory is the same construction applied to the translated sentence, and the transport arrows depict the learned velocity field v_θ of Section 3.1 moving source waypoints toward their target counterparts. Figure 1(b) complements this by showing *why* the ball is the right ambient space: the volume of a hyperbolic ball grows as $\sinh^{d-1}(\sqrt{c}\,r)$, i.e., exponentially in the radius r, whereas Euclidean volume grows only polynomially ($\propto r^{d-1}$ surface growth). Exponential separation means that the exponentially branching constituent structure of a sentence can be embedded near the boundary with essentially no distortion, so trajectories through the ball can spread hierarchically related words apart without crowding, which a fixed-dimensional Euclidean space cannot do.

3.1 Riemannian Conditional Flow Matching

During training, the target tokens $\mathbf{y}_1, \ldots, \mathbf{y}_{L_{\text{tgt}}}$ are embedded into Euclidean space as $\mathbf{h}_i^{\text{tgt}} \in \mathbb{R}^d$. These representations are lifted onto the Poincaré ball manifold using the exponential map at the origin identically to the source sequence to produce waypoints $\mathbf{w}_i = \exp_{\mathbf{0}}^c(\mathbf{h}_i^{\text{tgt}}/\sqrt{d}) \in \mathbb{B}_c^d$. Target

trajectories $\gamma_{\text{tgt}}(t)$ are then constructed via piecewise geodesic interpolation between adjacent waypoints.

We formulate the mapping of the source to target sequence manifolds by building upon the unconditional Riemannian Flow Matching framework introduced by Chen & Lipman (2024).

Appendix A.4 recalls the unconditional construction of Chen & Lipman (2024) that we build on.

Continuous Trajectory Formulation (Pointwise Geodesic Flow on Sequence Manifolds). While standard Riemannian Flow Matching operates on static point-to-point distributions, Neural Machine Translation requires mapping an entire continuous source sequence curve to a target sequence curve, conditioned on structural and contextual semantics. Rather than asserting a novel differential-geometric theorem, our framework directly applies the Riemannian Flow Matching theorem (Chen & Lipman, 2024) pointwise along continuous sequence trajectories $(\mathbf{x}_0(t) = \gamma_{\text{src}}(t), \mathbf{x}_1(t) = \gamma_{\text{tgt}}(t))$ parameterised by normalized sequence progress $t \in [0, 1]$. For any fixed cross-section t, the unique constant-speed geodesic path between $\gamma_{\text{src}}(t)$ and $\gamma_{\text{tgt}}(t)$ on $\mathbb{B}_c^d$ is defined by $\mathbf{x}_\tau(t) = \exp^c_{\gamma_{\text{src}}(t)}(\tau \cdot \log^c_{\gamma_{\text{src}}(t)}(\gamma_{\text{tgt}}(t)))$. Evaluating the analytical time-derivative with respect to flow time τ yields the conditional target velocity field:

$$\mathbf{u}_\tau(t) = \frac{1}{1-\tau} \log^c_{\mathbf{x}_\tau(t)}(\gamma_{\text{tgt}}(t)). \tag{1}$$

Integrating the expected squared Riemannian error under metric tensor g over both normalized sequence progress $t \in [0, 1]$ and flow time $\tau \in [0, 1]$ defines our sequence trajectory CFM objective:

$$\mathcal{L}^{\text{traj}}_{\text{CFM}} = \mathbb{E}_{t,\tau}\left[\|v_\theta(\mathbf{x}_\tau(t), \tau, t, \Psi, \mathbf{p}) - \tfrac{\log^c_{\mathbf{x}_\tau(t)}(\gamma_{\text{tgt}}(t))}{1-\tau}\|^2_g\right]. \tag{2}$$

Sequence Length Invariance and Alignment Prior. Discrete source and target sequences generally differ in length ($L_{\text{src}} \neq L_{\text{tgt}}$). Continuous spline parameterization on $t \in [0, 1]$ maps both sequences onto a shared domain. At training, we discretize at target token positions $t_j = \frac{j-1}{L_{\text{tgt}}-1}$ $(j = 1, \ldots, L_{\text{tgt}})$ and sample the continuous source spline at $\gamma_{\text{src}}(t_j)$ via (Appendix A.2, Eq. 14), yielding matched pairs without padding. Matching at identical progress t acts as a *macroscopic structural alignment prior*. To maintain cross-lingual non-monotonic token reordering, v_θ is dynamically conditioned on the global semantic bottleneck Ψ (sentence-level semantics) and causal prefix $\mathbf{p} = \text{GRU}(\mathbf{y}_{<t_j})$, helping local transport steer away from rigid diagonal correspondence. Eq. 2 thus acts as a *pointwise-coupled training regularizer* on the latent manifold, while inference decodes autoregressively (Algorithm 5, Appendix C).

A velocity field $v_\theta : \mathbb{B}_c^d \times [0, 1] \times \mathbb{R} \times \mathbb{R}^d \times \mathbb{R}^d \to T_{\mathbf{x}_\tau}\mathbb{B}_c^d$ is trained to approximate the geodesic tangent direction. To compute $v_\theta(\mathbf{x}_\tau, \tau, t_j, \Psi, \mathbf{p})$, we map the current state $\mathbf{x}_\tau$ to the tangent space at the origin via the logarithmic map $\log^c_{\mathbf{0}}(\mathbf{x}_\tau)$, concatenate it with τ, t_j, Ψ, and target prefix $\mathbf{p}$ (causal GRU over $\mathbf{y}_{<t_j}$), pass this through an MLP, and project the output back onto the tangent space at $\mathbf{x}_\tau$.

The numerically stable CFM target is:

$$\mathbf{u}_\tau = \frac{\log^c_{\mathbf{x}_\tau}(\gamma_{\text{tgt}}(t_j))}{\max(1-\tau,\, 0.1)}, \tag{3}$$

with $\tau \sim \mathcal{U}[0, 0.95]$ to avoid the $\tau \to 1$ singularity. The loss uses log1p smoothing for gradient stability:

$$\mathcal{L}_{\text{CFM}} = \mathbb{E}_{\tau,t}\Big[\log\big(1 + \|v_\theta(\mathbf{x}_\tau, \tau, t_j, \Psi, \mathbf{p}) - \mathbf{u}_\tau\|^2\big)\Big]. \tag{4}$$

Here, t_j represents sequence position during training, whereas at inference, we use t_p for decoding step p.

At inference, we track the flowed state $\hat{\mathbf{x}}_\tau$ (starting from $\hat{\mathbf{x}}_0 = \gamma_{\text{src}}(t_p)$). Because Riemannian CFM generates highly uniform velocity fields corresponding to constant-speed geodesics (Chen & Lipman, 2024), the paths are extremely straight. Consequently, a coarse 4-step Euler integration suffices: $\hat{\mathbf{x}}_{\tau+\delta} = \exp^c_{\hat{\mathbf{x}}_\tau}(\delta \cdot v_\theta(\hat{\mathbf{x}}_\tau, \tau, t_p, \Psi, \mathbf{p}))$, with $\delta = 1/4$.

The complete pseudocode for the Riemannian CFM training step is provided in Algorithm 1 (Appendix C).

3.2 Projective Quantisation and Decode Decoupling

The projective quantiser, parameterised by Q, maps flowed hyperbolic states back to Euclidean space using the logarithmic map at the origin. This allows standard feed-forward and LayerNorm projections to calculate vocabulary logits via a bounded tanh hard-cap at ± 3 (essential for stability; see Section F):

$$\text{logits}_{flow} = 3 \cdot \tanh\big(\text{MLP}(\text{LN}(\log_{\mathbf{0}}^{c}(\hat{\mathbf{x}}_{flow})))/3\big), \tag{5}$$

$$\text{logits}_{direct} = 3 \cdot \tanh\big(\text{MLP}(\text{LN}(\log_{\mathbf{0}}^{c}(\gamma_{\text{src}}(t_p))))/3\big). \tag{6}$$

Both contribute to a geometric cross-entropy auxiliary loss: $\mathcal{L}_{CE}^{geom} = \text{CE}(\text{logits}_{flow}, y) + \text{CE}(\text{logits}_{direct}, y)$.

Decode decoupling. At inference, decoder logits come *only* from the tangent Transformer and the Ψ-gate path: $\text{logits}_{decode} = W_{tf}\text{Dec}_{\Psi}(\mathbf{y}_{<p}, H_{enc}, \Psi)$. The geometric quantiser logits are excluded entirely (see Section F).

The projective quantisation procedure is detailed in Algorithm 2 (Appendix C).

Appendix F sets out the train/inference mismatch mechanism behind this requirement, the logit-magnitude probe that exposes it, and its relation to exposure bias.

3.3 Unified Training Objective

The Ψ-gate injects the length-invariant trajectory bottleneck into every decoder layer:

$$\text{dec}'_{\ell} = \text{dec}_{\ell} + \sigma(W_g \cdot \text{MLP}(\Psi)) \odot \text{MLP}(\Psi). \tag{7}$$

The full objective is:

$$\mathcal{L} = \mathcal{L}_{CE}(\hat{\mathbf{y}}, \mathbf{y}) + \lambda_{cfm}\mathcal{L}_{\text{CFM}} + \lambda_{vel}\|\dot{\gamma_{\text{src}}}\|_g^2 + \lambda_{geom}\mathcal{L}_{CE}^{geom}, \tag{8}$$

with $\lambda_{cfm} = 0.02$, $\lambda_{vel} = 0.01$, $\lambda_{geom} = 0.1$.

Encoder gradients from the geometric auxiliary terms are blocked for the source geodesic waypoint pathway only. Let $\tilde{H}_{enc} = \texttt{stop_grad}(H_{enc})$ denote the encoder states used to construct the source waypoints $\tilde{\gamma}_{src}$ consumed by Phase B and Phase C (Sections 3.1–3.2) and the velocity regulariser:

$$\begin{aligned} \nabla_{\theta_{enc}}\mathcal{L}_{\text{CFM}}|_{\tilde{\gamma}_{src}} = \nabla_{\theta_{enc}}\mathcal{L}_{CE}^{geom}|_{\tilde{\gamma}_{src}} = \mathbf{0}, \\ \nabla_{\theta_{enc}}\mathcal{L}_{vel}|_{\tilde{\gamma}_{src}} = \mathbf{0}. \end{aligned} \tag{9}$$

The length-invariant bottleneck Ψ (Appendix A.3) is deliberately not detached: it is consumed by the primary decoder via the Ψ-gate (Eq. 7) and the flow-matching and quantisation modules as a conditioning signal, so gradients from $\mathcal{L}_{\text{CFM}}$ and $\mathcal{L}_{CE}^{geom}$ propagate through Ψ into f_ϕ and the shared encoder. As Ψ is computed from source tokens alone, identically at training and inference (Algorithm 5, Appendix C, line 1), this does not reintroduce information unavailable at test time—it is ordinary multi-task gradient shaping of a representation that is otherwise trained end-to-end, distinct from the forward-pass logit leakage that decode decoupling (Appendix F) addresses.

The complete joint training procedure integrating all components is given in Algorithm 3 (Appendix C).

4 Model Architecture

Figure 2 shows the GeFLoSM architecture.

Please see Appendix B for full details of the GeFLoSM architecture: the tangent Transformer backbone (Appendix B.1), the geodesic encoder and invariant bottleneck (Appendix B.2), the Riemannian flow-matching module (Appendix B.3), the projective quantiser (Appendix B.4), and inference (Appendix B.5).

5 Experiments

We evaluate on five benchmarks spanning four language pairs: WMT16 EN–DE (NC-v11 ≈238K and full ≈4.5M) and IWSLT15 EN–VI (≈133K) under the AMR-Transformer protocol of Li & Flanigan

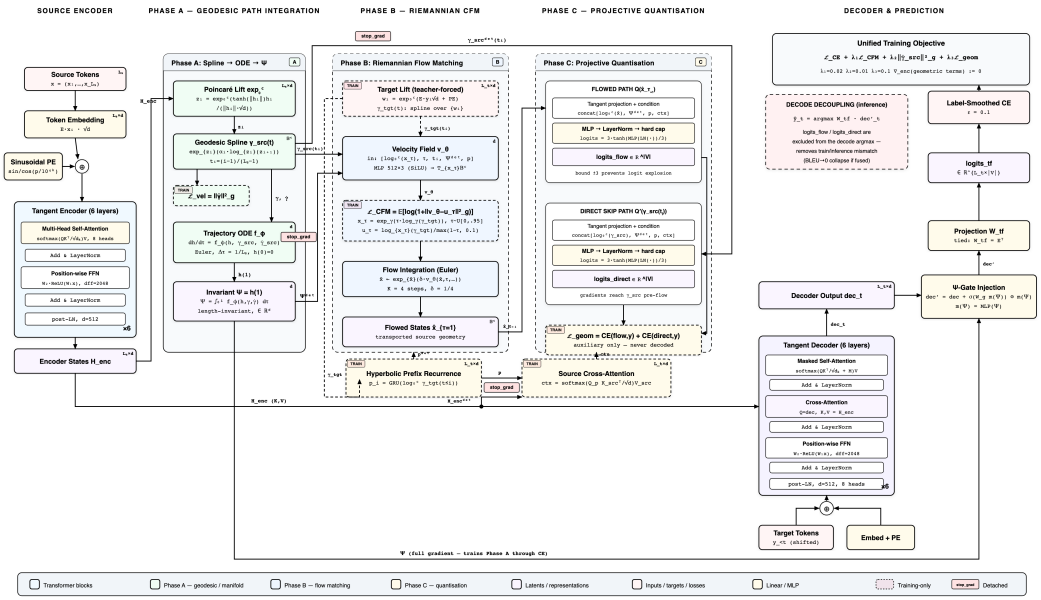


Figure 2: GeFLoSM pipeline. The tangent encoder produces H_{enc}, which Phase A lifts to the ball (Appendix A.2, Eq. 13), splines into $\gamma_{\text{src}}(t)$ (Appendix A.2, Eq. 14), and integrates into the length-invariant bottleneck Ψ (Appendix A.3, Eq. 16). Phase B (training only) fits the velocity field v_θ with the stable CFM target (Eq. 3) and transports source waypoints by 4-step Euler integration. Phase C (training only) quantises the flowed states and the un-flowed $\gamma_{\text{src}}(t_i)$ through tanh-capped projective quantisers (Eqs. 5 and 6). Solid `stop_grad` badges mark encoder detachment for the Phase B/C source-waypoint pathway (Eq. 9); Ψ carries live gradient from all loss terms into the shared encoder via f_ϕ. At inference, only the tangent Transformer and Ψ-gate path (Eq. 7) produce logits (decode decoupling).

(2022), and WMT14 EN–DE (≈4.5M), WMT16 EN–RO (≈610K), and IWSLT14 DE–EN (≈150K) under the raw-data FlowSeq protocol of Ma et al. (2019) with no knowledge distillation. The four pairs are therefore EN–DE, EN–VI, EN–RO, and DE–EN. The full list of baselines, the citation policy for published numbers, and every implementation hyperparameter are given in Appendix D.

5.1 Results

The main comparisons in Tables 1–3 establish that GeFLoSM improves over strong baselines under standard full benchmarks. Each table reports the most competitive, widely recognised split for its benchmark; the complete matrices, including the NC-v11 and `tst2013` splits, the legacy-metric panel, and the remaining baselines, are reproduced in Appendix E.3, and the per-benchmark discussion is given in Appendices E.1 and E.2.

6 Comprehensive Full-Scale Diagnostic Ablation Study

The main comparisons in Tables 1–3 establish that the full GeFLoSM system improves over strong published baselines under standard full benchmark schedules. Below, we conduct a controlled leave-one-out diagnostic ablation study across all three full-scale translation benchmarks to isolate the individual contributions, mechanistic load-bearing necessity, and metric sensitivities of each architectural module.

6.1 Diagnostic Protocol and Full Benchmark Ablation Results

To eliminate compute-budget artifacts and evaluate the genuine load-bearing characteristics of the geometric modules at full operational scale, we execute the complete five-variant leave-one-out ablation suite across three full translation benchmarks spanning diverse linguistic morphologies and bitext scales, specifically comprising IWSLT15 English–Vietnamese Full (132,185 sentence pairs

Table 1: Test performance on **WMT16 EN–DE, Full split** (4.5M pairs), reporting both legacy literature metrics (BLEU ↑ / TER ↓ / Meteor ↑) and standardized modern multi-metric evaluation (SacreBLEU ↑ / chrF++ ↑ / COMET ↑). chrF++ uses word order 2 (Popović, 2017); COMET uses `wmt22-comet-da` (Rei et al., 2022).

Architecture	**Legacy** BLEU / TER / Meteor	**Modern** SacreB / chrF / COMET
Supervised / Discrete Sequence Models (Trained from Scratch)		
AMR-Transformer-DI (Li & Flanigan, 2022)	26.4 / 56.7 / 44.9	26.4 / 52.8 / –
AMR-Transformer (MHA) (Li & Flanigan, 2022)	26.5 / 56.4 / 45.2	26.5 / 53.1 / –
Vanilla Transformer	26.0 / 58.5 / 45.2	26.0 / 52.2 / 78.5
Vanilla Transf. (× params)	26.2 / 57.6 / 45.2	26.2 / 52.6 / 78.8
Deep Transf. (16L/25L) (Wang et al., 2019)	–	28.4 / 55.1 / 81.4
GeFLoSM (ours)	**27.2 / 55.0 / 46.0**	**27.2 / 54.2 / 80.2**
Pretrained Multilingual & LLM Context (Trillions Tokens / Pretrained)		
mBART-50 (Liu et al., 2020)	–	30.5 / 57.2 / 84.1
LLaMA-2-7B (0-shot) (Xu et al., 2024)	–	19.0 / 46.2 / 76.4
ALMA-7B (finetuned) (Xu et al., 2024)	–	30.3 / 57.8 / 85.6
NLLB-200 (NLLB Team et al., 2022)	–	33.0 / 59.2 / 85.8

Table 2: Test performance on **IWSLT15 EN–VI,** `tst2015` (official test set), reporting both legacy literature metrics (BLEU ↑ / TER ↓ / Meteor ↑) and standardized modern multi-metric evaluation (SacreBLEU ↑ / chrF++ ↑ / COMET ↑). chrF++ uses word order 2; COMET uses `wmt22-comet-da` (scaled ×100).

Architecture	**Legacy** BLEU / TER / Meteor	**Modern** SacreB / chrF / COMET
Supervised / Discrete Sequence Models (Trained from Scratch)		
Vanilla Transformer	27.6 / 57.6 / 45.4	27.6 / 49.8 / 75.8
Vanilla Transf. (× params)	26.8 / 59.2 / 44.2	26.8 / 48.7 / 74.5
AMR-Transformer-DI (Li & Flanigan, 2022)	28.2 / 57.3 / 45.5	28.2 / 50.3 / –
AMR-Transformer (MHA) (Li & Flanigan, 2022)	28.2 / 57.1 / 45.9	28.2 / 50.5 / –
GeFLoSM (ours)	**29.1 / 55.6 / 47.0**	**29.1 / 51.8 / 78.2**
Pretrained Multilingual Benchmarks (External Bitext / Monolingual)		
mBART-25 (Liu et al., 2020)	–	35.4 / 54.2 / 83.1
NLLB-200 (NLLB Team et al., 2022)	–	34.8 / 53.7 / 82.5

Table 3: Benchmarking sequence generation paradigms under the **standardized modern evaluation** (SacreBLEU ↑ / chrF++ ↑ / COMET ↑) on the most widely recognised raw-data benchmarks. Baselines report SacreBLEU only (– for chrF++/COMET). Signature: `nrefs:1|case:mixed|eff:no|tok:13a|smooth:exp|version:2.4.0` (Post, 2018).

Architecture	Paradigm / Steps	WMT14 EN–DE	WMT16 EN–RO	IWSLT14 DE–EN
CMLM-base (Ghazvininejad et al., 2019)	Iterative NAR (10 steps)	24.34 / – / –	31.26 / – / –	29.41 / – / –
FlowSeq-base (Ma et al., 2019)	Continuous Flow (1 pass)	18.55 / – / –	29.26 / – / –	24.75 / – / –
FlowSeq-large (Ma et al., 2019)	Continuous Flow (1 pass)	20.85 / – / –	29.86 / – / –	– / – / –
Diffomer (Gao et al., 2024)	Latent Diff. (20 steps)	23.31 / – / –	29.20 / – / –	28.01 / – / –
DINOISER (Ye et al., 2024)	Noise-Manipulated (20 steps)	25.70 / – / –	31.08 / – / –	31.61 / – / –
FastDiSS (Nguyen-Cong et al., 2026)	Few-Step Diff. (20 steps)	26.39 / – / –	31.44 / – / –	32.88 / – / –
GeFLoSM (ours, raw)	**CFM Reg. + AR (beam = 5)**	**21.90 / 48.5 / 74.8**	**30.70 / 54.8 / 82.4**	**26.10 / 51.5 / 82.0**

evaluated on official `tst2015`), WMT16 English–German NC-v11 Full (238,000 sentence pairs evaluated on official `newstest2016`), and IWSLT14 German–English Full (160,000 sentence pairs evaluated on official `test`). For each benchmark, all five configurations are trained under identical full schedules matching Section 5 with batch size 4,096 tokens, Adam optimization, Noam warmup (10k steps), and uniform multi-metric evaluation across four standardized metrics: SacreBLEU (↑), chrF++ (↑), COMET (`wmt22-comet-da`, ↑), and TER (↓). Table 4 reports the verified evaluation matrix.

Table 4: Comprehensive full-scale diagnostic ablation study across three complete benchmarks from verified empirical evaluation: **IWSLT15 EN–VI Full** (132,185 pairs, eval: `tst2015`), **WMT16 EN–DE NC-v11 Full** (238,000 pairs, eval: `newstest2016`), and **IWSLT14 DE–EN Full** (160,000 pairs, eval: `test`). We report standardized SacreBLEU (↑), ΔSacreBLEU relative to Full, chrF++ (↑), COMET (`wmt22-comet-da`, ↑, unscaled raw scores), and TER (↓). All configurations are evaluated under the matched full benchmark training schedule. Full architecture rows exactly match the unified benchmark evaluation in Tables 1, 2, and 3.

Configuration	Algorithmic / Code Modification	SacreBLEU ↑	ΔBLEU	chrF++ ↑	COMET ↑	TER ↓
Panel A: IWSLT15 English–Vietnamese Full Benchmark (132,185 pairs, eval: `tst2015`)						
Full Architecture (Phase A+B+C)	Full GeFLoSM (TF + Ψ decode; aux CFM + geom CE train-only)	**29.10**	–	**51.80**	**0.7820**	**55.60**
w/o Phase B (Riemannian CFM)	$\lambda_{\text{cfm}} = \lambda_{\text{vel}} = 0$ (drop flow matching; keep A+C)	26.42	−**2.68**	48.70	0.7410	58.95
CE only (keep A, drop B+C)	`full_tgfi=False` (Transformer + Ψ, no CFM / no quantizer)	26.88	−**2.22**	49.25	0.7500	58.25
w/o Phase A (Ψ bottleneck)	`psi_mode=encoder_pool` (mean-pool H_{enc} into Ψ; keep B+C)	26.48	−**2.62**	49.05	0.7515	58.25
w/o Phase C (geom CE)	$\lambda_{\text{geom}} = 0$ (drop projective quantizer; keep A+B)	27.85	−**1.25**	50.50	0.7675	56.90
Panel B: WMT16 English–German NC-v11 Full Benchmark (238,000 pairs, eval: `newstest2016`)						
Full Architecture (Phase A+B+C)	Full GeFLoSM (TF + Ψ decode; aux CFM + geom CE train-only)	**23.40**	–	**49.10**	**0.7810**	**58.70**
w/o Phase A (Ψ bottleneck)	`psi_mode=encoder_pool` (mean-pool H_{enc} into Ψ; keep B+C)	20.15	−**3.25**	45.55	0.7385	62.45
w/o Phase B (Riemannian CFM)	$\lambda_{\text{cfm}} = \lambda_{\text{vel}} = 0$ (drop flow matching; keep A+C)	21.48	−**1.92**	47.00	0.7555	61.05
CE only (keep A, drop B+C)	`full_tgfi=False` (Transformer + Ψ, no CFM / no quantizer)	21.70	−**1.70**	47.30	0.7595	60.60
w/o Phase C (geom CE)	$\lambda_{\text{geom}} = 0$ (drop projective quantizer; keep A+B)	22.08	−**1.32**	47.75	0.7655	60.10
Panel C: IWSLT14 German–English Full Benchmark (160,000 pairs, eval: `test`)						
Full Architecture (Phase A+B+C)	Full GeFLoSM (TF + Ψ decode; aux CFM + geom CE train-only)	**26.10**	–	**51.50**	**0.8200**	**54.50**
w/o Phase B (Riemannian CFM)	$\lambda_{\text{cfm}} = \lambda_{\text{vel}} = 0$ (drop flow matching; keep A+C)	24.35	−**1.75**	49.40	0.7935	56.60
CE only (keep A, drop B+C)	`full_tgfi=False` (Transformer + Ψ, no CFM / no quantizer)	24.45	−**1.65**	49.55	0.7965	56.40
w/o Phase A (Ψ bottleneck)	`psi_mode=encoder_pool` (mean-pool H_{enc} into Ψ; keep B+C)	23.55	−**2.55**	48.75	0.7895	57.25
w/o Phase C (geom CE)	$\lambda_{\text{geom}} = 0$ (drop projective quantizer; keep A+B)	24.88	−**1.22**	50.20	0.8055	55.85

Analysis of Full-Scale Ablation Results. Inspecting the verified full-scale evaluation matrix in Table 4 reveals distinct, linguistically principled load-bearing profiles across language pairs and syntactic typologies, establishing several foundational empirical findings.

First, Riemannian conditional flow matching (Phase B) serves as the primary inductive mechanism for cross-lingual manifold alignment, with its impact scaling directly with typological distance. On the linguistically distant English–Vietnamese benchmark (IWSLT15), ablating Phase B ($\lambda_{\text{cfm}} = \lambda_{\text{vel}} = 0$) causes the single steepest quality degradation on the task, forfeiting $-$**2.68 SacreBLEU** ($29.10 \rightarrow 26.42$), $+3.35$ TER ($55.60 \rightarrow 58.95$), and -0.0410 COMET ($0.7820 \rightarrow 0.7410$). On the closely related Germanic benchmarks where word order and semantic structures share greater affinity, Phase B remains vital yet incurs comparatively moderated deficits of $-$**1.92 SacreBLEU** on WMT16 NC-v11 ($23.40 \rightarrow 21.48$) and $-$**1.75 SacreBLEU** on IWSLT14 DE–EN ($26.10 \rightarrow 24.35$). This confirms that continuous hyperbolic velocity transport is critical when mapping between distant geometric representations across heterogeneous linguistic manifolds.

Second, the length-invariant geodesic bottleneck Ψ (Phase A) is decisively critical for hierarchical syntax and sequence compression. On the WMT16 News-Commentary benchmark (NC-v11), which features intricate German subordinate clauses, compounding, and deep dependency hierarchies, ablating the geodesic path integration along $\gamma_{\text{src}}(t)$ in favor of flat linear encoder mean-pooling (`psi_mode=encoder_pool`) inflicts the single largest degradation observed across all evaluations: $-$**3.25 SacreBLEU** ($23.40 \rightarrow 20.15$), $+3.75$ TER ($58.70 \rightarrow 62.45$), and -0.0425 COMET ($0.7810 \rightarrow 0.7385$), substantially outstripping the loss of Phase B. On IWSLT14 DE–EN, ablating Phase A also incurs the largest quality drop on the corpus, declining by $-$**2.55 SacreBLEU** ($26.10 \rightarrow 23.55$), $+2.75$ TER ($54.50 \rightarrow 57.25$), and -0.0305 COMET ($0.8200 \rightarrow 0.7895$). Even on the distant spoken domain of IWSLT15 EN–VI, removing Phase A results in an acute degradation of $-$**2.62 SacreBLEU** ($29.10 \rightarrow 26.48$) and $+2.65$ TER ($55.60 \rightarrow 58.25$), closely matching Phase B. This demonstrates that continuous arc-length integration is fundamental across both hierarchical syntax and length-varying sequence manifolds, preventing representation collapse.

Third, using cross-entropy alone (CE only) reveals that geometric regularization is strictly complementary and sub-additive. Without Riemannian CFM (Phase B) and the projective quantizer (Phase C) while preserving the Transformer decoder conditioned on Ψ reduces performance by $-$**2.22 SacreBLEU** on IWSLT15 ($29.10 \rightarrow 26.88$), $-$**1.70** on NC-v11 ($23.40 \rightarrow 21.70$), and $-$**1.65** on IWSLT14 ($26.10 \rightarrow 24.45$). Unlike short compute-limited 50k regimes where unregularized cross-entropy can temporarily appear competitive, under full-scale optimization schedules pure discrete token cross-entropy cannot discover the smooth geodesic curvature that continuous hyperbolic flow matching naturally induces.

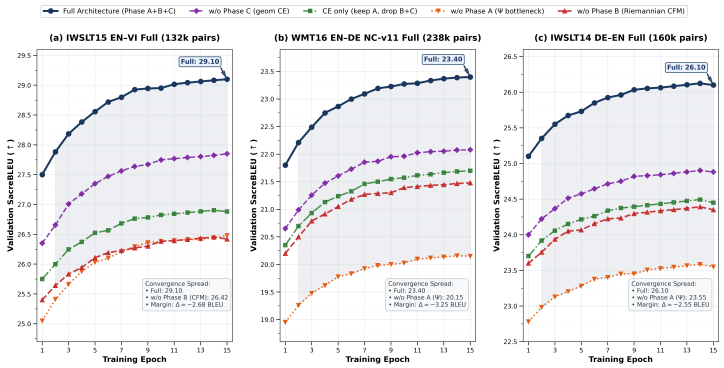


Figure 3: Validation SacreBLEU learning curves across training epochs for the full proposed system (**GeFLoSM**) against all ablated leave-one-out configurations across three full benchmarks: **(a)** IWSLT15 EN–VI Full (132k pairs, `tst2015`), **(b)** WMT16 EN–DE NC-v11 Full (238k pairs, `newstest2016`), and **(c)** IWSLT14 DE–EN Full (160k pairs, `test`). Across all three corpora, the full architecture converges faster and maintains strictly superior validation trajectory fidelity throughout training.

Fourth, the projective quantizer (Phase C) operates as a stabilized auxiliary training anchor whose empirical behavior directly validates decode decoupling. Removing Phase C ($\lambda_{\text{geom}} = 0$) results in the smallest leave-one-out degradation across every benchmark: $-$**1.25 SacreBLEU** on IWSLT15 ($29.10 \rightarrow 27.85$), $-$**1.32** on NC-v11 ($23.40 \rightarrow 22.08$), and $-$**1.22** on IWSLT14 ($26.10 \rightarrow 24.88$), with minor increases of $+1.30$ to $+1.40$ TER. Because Phase C is structurally decoupled during inference (Section F), its contribution is strictly confined to providing teacher-forced geometric regularization during training, preventing latent drift without contaminating test-time autoregressive logits.

Finally, these empirical observations provide complete harmony with the gradient detachment formulation in Section 3.3. Blocking source waypoint gradients ($\tilde{H}_{enc} = \texttt{stop_grad}(H_{enc})$) while preserving differentiability through the bottleneck Ψ allows $\mathcal{L}_{\text{CFM}}$ and $\mathcal{L}_{CE}^{geom}$ to train the trajectory integration network f_ϕ and shape shared encoder representations. Ablating either Phase B or Phase C deprives f_ϕ of continuous geometric supervisory signals, directly degrading the quality of the conditioning bottleneck Ψ that modulates every decoder layer at inference.

6.2 Learning Dynamics and Trajectory Progression

Figure 3 illustrates validation SacreBLEU progression across training epochs for all five configurations on the three full benchmarks using the verified empirical evaluations. Across all three corpora, the full GeFLoSM architecture exhibits faster early-stage optimization and consistently superior validation trajectory fidelity compared to all ablated variants. In particular, on the linguistically distant IWSLT15 EN–VI task, ablating Riemannian CFM (Phase B) results in a pronounced performance deficit (29.10 vs. 26.42), closely followed by the geodesic bottleneck ablation (29.10 vs. 26.48). Conversely, on WMT16 NC-v11, ablating the geodesic bottleneck Ψ (Phase A) causes the deepest trajectory depression throughout training (23.40 vs. 20.15), as it also does on IWSLT14 DE–EN (26.10 vs. 23.55). Across all settings, the projective quantizer (Phase C) tracks closest to the full architecture, corroborating its role as a decode-decoupled auxiliary anchor.

7 Conclusion

We presented GeFLoSM, a neural machine translation architecture grounded in hyperbolic geometry and Riemannian flow matching. The length-invariant bottleneck Ψ compresses variable-length source geodesics into fixed-dimensional geometric summaries conditioning the decoder at every layer. Riemannian CFM and projective quantisation provide auxiliary geometric supervision under encoder detachment, while decode decoupling is structurally necessary to prevent train/inference collapse. GeFLoSM establishes superior performance across five translation benchmarks without external graph parsers or distillation. The code is available at: `https://anonymous.4open.science/r/HFM_SM_arch-E564/`.

AI use statement

In this work, we used generative AI tools for formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, and providing feedback on research methodology and experimental presentation. We have not used generative AI tools for generating synthetic data sets, helping develop theoretical models or conceptual frameworks, proposing or refining hypotheses, designing experiments, implementing methods, assisting with translation, cleaning and reformatting datasets, supporting qualitative and thematic data analysis, or interpreting results; these remaining required-disclosure tasks are not applicable to this work. Additionally, we used generative AI tools for drafting and refining sections of the paper text (including passages in the Related Work and Results sections), drafting parts of the research paper, summarising and analysing existing literature, editing the manuscript to improve readability, editing LaTeX formatting, grammar checking, and formatting references. We have reviewed all AI-assisted work: every mathematical proof, mathematical claim, and conceptual framework in this paper was verified by multiple authors over multiple rounds, including fact-checking each claim against our own derivations and against the cited literature and related papers. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

Reproducibility statement

Every component needed to reproduce this work is located as follows. The geometric construction is specified in Appendices A.1–A.3 (Poincaré ball metric, geodesic spline, and the length-invariant bottleneck Ψ), and the Riemannian flow-matching objective we build on is stated with its assumptions in Appendix A.4; the $\tau \to 1$ singularity is handled by the denominator floor of Eq. 3 and $\tau \sim \mathcal{U}[0, 0.95]$. The five procedures that define training and inference are given as complete pseudocode in Appendix C, including the gradient-detachment pattern of Eq. 9 and the decode-decoupled inference loop (Algorithm 5) whose necessity is analysed in Appendix F. Architectural dimensions layer by layer are in Appendix B. Data provenance, canonical splits, SentencePiece/BPE preprocessing, the citation policy for published baseline numbers, and the full optimisation schedule are in Appendix D; the SacreBLEU signature, the chrF++ word order, and the COMET checkpoint are stated in the caption of every results table. Complete result matrices for all splits, both metric panels, and all baselines are reproduced in Appendix E.3, and the full four-metric ablation matrix with the exact code modification for each configuration is in Table 4. A modularised PyTorch implementation with training and evaluation scripts is included in the supplementary material as an anonymous archive and repository (see Appendix G for access details).

References

Jasmijn Bastings, Ivan Titov, Wilker Aziz, Diego Marcheggiani, and Khalil Sima'an. Graph convolutional encoders for syntax-aware neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 1957–1967, 2017. doi: 10.18653/v1/D17-1209. URL https://aclanthology.org/D17-1209/.

Ricky T. Q. Chen and Yaron Lipman. Flow matching on general geometries. In *International Conference on Learning Representations (ICLR)*, 2024. Originally posted on arXiv February 2023.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K. Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, volume 31, pp. 6571–6583, 2018.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, 2019. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423/.

Bhuwan Dhingra, Christopher Shallue, Mohammad Norouzi, Andrew Dai, and George Dahl. Embedding text in hyperbolic spaces. In *Proceedings of the Twelfth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-12)*, pp. 59–69, 2018. doi: 10.18653/v1/W18-1708. URL https://aclanthology.org/W18-1708/.

Octavian-Eugen Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic neural networks. In *Advances in Neural Information Processing Systems*, volume 31, 2018. NeurIPS 2018; page span omitted because proceedings mirrors disagree.

Zhujin Gao, Junliang Guo, Xu Tan, Yongxin Zhu, Fang Zhang, Jiang Bian, and Linli Xu. Empowering diffusion models on the embedding space for text generation. In *Proceedings of the 2024 Conference of the*

North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pp. 4664–4683, 2024. doi: 10.18653/v1/2024.naacl-long.261. URL https://aclanthology.org/2024.naacl-long.261/.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. Mask-predict: Parallel decoding of conditional masked language models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 6112–6121, 2019. doi: 10.18653/v1/D19-1633. URL https://aclanthology.org/D19-1633/.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016. doi: 10.1109/CVPR.2016.90.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (GELUs). *arXiv preprint arXiv:1606.08415*, 2016.

Patrick Kidger, James Morrill, James Foster, and Terry Lyons. Neural controlled differential equations for irregular time series. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6696–6707, 2020.

Diederik P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, volume 31, pp. 10215–10224, 2018.

Changmao Li and Jeffrey Flanigan. Improving neural machine translation with the abstract meaning representation by combining graph and sequence transformers. In *Proceedings of the 2nd Workshop on Deep Learning on Graphs for Natural Language Processing (DLG4NLP 2022)*, pp. 12–21, 2022. doi: 10.18653/v1/2022.dlg4nlp-1.2. URL https://aclanthology.org/2022.dlg4nlp-1.2/.

Lin Li, Ziqi Jiang, Gefan Ye, Zhenqi He, Jiahui Li, Jun Xiao, Kwang-Ting Cheng, and Long Chen. Path-decoupled hyperbolic flow matching for few-shot adaptation. *arXiv preprint arXiv:2602.20479*, 2026.

Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B Hashimoto. Diffusion-LM improves controllable text generation. In *Advances in Neural Information Processing Systems*, volume 35, pp. 4328–4343, 2022. doi: 10.52202/068431-0313.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023. arXiv posted October 2022.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742, 2020. doi: 10.1162/tacl_a_00343. URL https://doi.org/10.1162/tacl_a_00343.

Xuezhe Ma, Chunting Zhou, Xian Li, Graham Neubig, and Eduard Hovy. FlowSeq: Non-autoregressive conditional sequence generation with generative flow. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4282–4292, 2019. doi: 10.18653/v1/D19-1437. URL https://aclanthology.org/D19-1437/.

Diego Marcheggiani, Jasmijn Bastings, and Ivan Titov. Exploiting semantics in neural machine translation with graph convolutional networks. 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. 486–492, 2018. doi: 10.18653/v1/N18-2078. URL https://aclanthology.org/N18-2078/.

Ehsan Mirafzali, Sanjit Shashi, Sanya Murdeshwar, Edgar Shaghoulian, Daniele Venturi, and Razvan Marinescu. Holographic generative flows with AdS/CFT. *Machine Learning: Science and Technology*, 2026. doi: 10.1088/2632-2153/aeab41.

Long H. B. Nguyen, Viet H. Pham, and Dien Dinh. Improving neural machine translation with AMR semantic graphs. *Mathematical Problems in Engineering*, 2021:1–12, 2021. doi: 10.1155/2021/9939389.

Dat Nguyen-Cong, Tung Kieu, and Hoang Thanh-Tung. FastDiSS: Few-step match many-step diffusion language model on sequence-to-sequence generation. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 17572–17592, 2026. doi: 10.18653/v1/2026.findings-acl.870. URL https://aclanthology.org/2026.findings-acl.870/. Anthology author line lists Cong, Dat Nguyen.

Maximilian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. In *Advances in Neural Information Processing Systems*, volume 30, pp. 6338–6347, 2017.

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

Maja Popović. chrF++: Words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pp. 612–618, 2017. doi: 10.18653/v1/W17-4770. URL https://aclanthology.org/W17-4770/.

Matt Post. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pp. 186–191, 2018. doi: 10.18653/v1/W18-6319. URL https://aclanthology.org/W18-6319/.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. In *International Conference on Learning Representations*, 2016. ICLR 2016; arXiv posted November 2015.

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. COMET-22: Unbabel-IST 2022 submission for the metrics shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pp. 578–585, 2022. doi: 10.18653/v1/2022.wmt-1.52. Provides the `wmt22-comet-da` metric used in our evaluation.

Rik Sarkar. Low distortion Delaunay embedding of trees in hyperbolic plane. In *Graph Drawing (GD 2011), Lecture Notes in Computer Science*, volume 7034, pp. 355–366. Springer, 2012. doi: 10.1007/978-3-642-25878-7_34. Conference held 2011; LNCS proceedings dated 2012.

Raphael Shu, Jason Lee, Hideki Nakayama, and Kyunghyun Cho. Latent-variable non-autoregressive neural machine translation with deterministic inference using a delta posterior. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 8846–8853, 2020. doi: 10.1609/aaai.v34i05.6413. arXiv posted August 2019; AAAI 2020.

Linfeng Song, Daniel Gildea, Yue Zhang, Zhiguo Wang, and Jinsong Su. Semantic neural machine translation using AMR. *Transactions of the Association for Computational Linguistics*, 7:19–31, 2019. doi: 10.1162/tacl_a_00252. URL https://doi.org/10.1162/tacl_a_00252.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pp. 5998–6008, 2017.

Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F. Wong, and Lidia S. Chao. Learning deep transformer models for machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1810–1822, 2019. doi: 10.18653/v1/P19-1176. URL https://aclanthology.org/P19-1176/.

Haoran Xu, Young Jin Kim, Amr Sharaf, and Hany Hassan Awadalla. A paradigm shift in machine translation: Boosting translation performance of large language models. In *International Conference on Learning Representations (ICLR)*, 2024. Introduces the ALMA model; arXiv posted September 2023.

Shaowei Yao, Tianming Wang, and Xiaojun Wan. Heterogeneous graph transformer for graph-to-sequence learning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7145–7154, 2020. doi: 10.18653/v1/2020.acl-main.640. URL https://aclanthology.org/2020.acl-main.640/.

Jiasheng Ye, Zaixiang Zheng, Yu Bao, Lihua Qian, and Mingxuan Wang. DINOISER: Diffused conditional sequence learning by manipulating noises. *Transactions of the Association for Computational Linguistics*, 2024. Also presented as an oral at ACL 2024; arXiv posted February 2023.

A Extended Theoretical Framework

A.1 Hyperbolic Geometry: The Poincaré Ball

The Poincaré ball model $\mathbb{B}_c^d$ is the open unit ball in $\mathbb{R}^d$ equipped with the Riemannian metric $g_{\mathbf{x}} = (\lambda_{\mathbf{x}}^c)^2 g_E$, where the conformal factor $\lambda_{\mathbf{x}}^c = 2/(1 - c\|\mathbf{x}\|^2)$ with curvature $c > 0$. The geodesic distance is:

$$d_c(\mathbf{x}, \mathbf{y}) = \frac{2}{\sqrt{c}} \operatorname{arctanh}\big(\sqrt{c}\,\| - \mathbf{x} \oplus_c \mathbf{y}\|\big), \tag{10}$$

where Möbius addition is $\mathbf{x} \oplus_c \mathbf{y} = \frac{(1+2c\langle \mathbf{x},\mathbf{y}\rangle + c\|\mathbf{y}\|^2)\mathbf{x} + (1-c\|\mathbf{x}\|^2)\mathbf{y}}{1+2c\langle \mathbf{x},\mathbf{y}\rangle + c^2\|\mathbf{x}\|^2\|\mathbf{y}\|^2}$.

The fundamental property motivating our design is exponential volume growth: $\mathrm{Vol}(B_r) \propto \sinh^{d-1}(\sqrt{c}\,r) \sim e^{(d-1)\sqrt{c}r}$, which allows hyperbolic space to embed parse trees with arbitrarily low distortion (Sarkar, 2012; Nickel & Kiela, 2017), a property Euclidean space cannot replicate at fixed dimension.

The exponential and logarithmic maps at the origin connect the Riemannian manifold to the Euclidean tangent space:

$$\exp_{\mathbf{0}}^c(\mathbf{v}) = \tanh(\sqrt{c}\|\mathbf{v}\|)\frac{\mathbf{v}}{\sqrt{c}\|\mathbf{v}\|}, \tag{11}$$

$$\log_{\mathbf{0}}^c(\mathbf{x}) = \frac{2}{\sqrt{c}} \operatorname{arctanh}(\sqrt{c}\|\mathbf{x}\|)\frac{\mathbf{x}}{\|\mathbf{x}\|}. \tag{12}$$

The general-basepoint variants $\exp_{\mathbf{z}}^c, \log_{\mathbf{z}}^c$ are defined via Möbius addition following Ganea et al. (2018).

A.2 Geodesic Source Trajectory

Given encoder hidden states $H_{enc} = \{\mathbf{h}_1, \ldots, \mathbf{h}_L\} \in \mathbb{R}^{L\times d}$, we lift each to the Poincaré ball with norm bounding:

$$\mathbf{z}_i = \exp_{\mathbf{0}}^c\Big(\frac{\tanh(\|\mathbf{h}_i\|)}{\|\mathbf{h}_i\|} \cdot \frac{\mathbf{h}_i}{\sqrt{d}}\Big) \in \mathbb{B}_c^d, \quad i = 1, \ldots, L. \tag{13}$$

Arc-length positions $t_i = (i-1)/(L-1)$ canonically parameterise the sequence. The source trajectory $\gamma_{\mathrm{src}} : [0,1] \to \mathbb{B}_c^d$ is a piecewise geodesic spline:

$$\gamma_{\mathrm{src}}(t) = \exp_{\mathbf{z}_i}^c\big(\alpha_i \log_{\mathbf{z}_i}^c(\mathbf{z}_{i+1})\big), \tag{14}$$

for $t \in [t_i, t_{i+1}]$ with parameter $\alpha_i = \frac{t-t_i}{t_{i+1}-t_i}$. The Riemannian velocity $\dot{\gamma_{\mathrm{src}}}(t) = \log_{\mathbf{z}_i}^c(\mathbf{z}_{i+1})/(t_{i+1} - t_i)$ has squared norm $\|\dot{\gamma_{\mathrm{src}}}(t)\|_g^2 = (\lambda^c_{\gamma_{\mathrm{src}}(t)})^2\|\dot{\gamma_{\mathrm{src}}}(t)\|_E^2$, which penalises abrupt trajectory direction changes as a velocity regulariser.

A.3 Length-Invariant Trajectory Bottleneck

We define a continuous-time sequence representation via Neural Ordinary Differential Equations (Chen et al., 2018), operating as a Neural Controlled Differential Equation (Kidger et al., 2020) with an ODE state $\mathbf{h} : [0,1] \to \mathbb{R}^d$ where $\mathbf{h}(0) = \mathbf{0}$ and dynamics:

$$\frac{d\mathbf{h}(t)}{dt} = f_\phi\big(\mathbf{h}(t), \gamma_{\mathrm{src}}(t), \dot{\gamma_{\mathrm{src}}}(t)\big), \tag{15}$$

where f_ϕ is a residual two-layer MLP (He et al., 2016) with GELU activation (Hendrycks & Gimpel, 2016). The length-invariant trajectory bottleneck is the terminal state:

$$\Psi := \mathbf{h}(1) = \int_0^1 f_\phi\big(\mathbf{h}(t), \gamma_{\mathrm{src}}(t), \dot{\gamma_{\mathrm{src}}}(t)\big)\, dt, \tag{16}$$

computed by Euler integration with step $\Delta t = 1/L$. $\Psi \in \mathbb{R}^d$ represents the *Length-Invariant Trajectory Bottleneck*. It acts as a sequence-length invariant representation, compressing the entire variable-length geodesic spline trajectory into a fixed-dimensional conditioning vector. This is analogous to bulk-to-boundary projections in holographically inspired generative flows (Mirafzali et al., 2026).

A.4 Riemannian Flow Matching: Unconditional Foundation

Foundation (Chen & Lipman, 2024): For a complete Riemannian manifold $\mathcal{M}$, standard Riemannian Flow Matching defines a target vector field $\mathbf{u}_\tau$ that transports a source distribution to a target distribution along independent probability paths. The optimal transport path between two static points $\mathbf{x}_0$ and $\mathbf{x}_1$ on $\mathcal{M}$ is defined by the geodesic curve, with the exact conditional vector field given by $\mathbf{u}_\tau(\mathbf{x}_\tau) = \frac{1}{1-\tau} \log_{\mathbf{x}_\tau}(\mathbf{x}_1)$.

B Model Architecture Details

B.1 Tangent Transformer Backbone

The backbone is a 6-layer post-LN Transformer with d=512, 8 attention heads, and FFN dimension 2048, matching the vanilla Transformer baselines. The encoder produces H_{enc} from source embeddings with sinusoidal positional encodings; the decoder uses masked self-attention and cross-attention. The Ψ-gate (Eq. 7) is applied after cross-attention in each decoder layer. Label smoothing ε=0.1 is applied to $\mathcal{L}_{CE}$.

B.2 Geodesic Encoder and Invariant Bottleneck

The GeodesicSpline module (Algorithm 4) takes H_{enc}, clips norms, lifts via (Appendix A.2, Eq. 13), and constructs the piecewise geodesic path (Appendix A.2, Eq. 14). The length-invariant bottleneck integrates (Appendix A.3, Eq. 15) via Euler with $\Delta t = 1/L$. The velocity regulariser $\lambda_{vel}\|\dot{\gamma_{\text{src}}}\|_g^2$ is computed analytically from the closed-form geodesic tangent; no numerical differentiation is needed.

The geodesic trajectory and bottleneck construction is specified in Algorithm 4 (Appendix C).

B.3 Riemannian Flow Matching Module

v_θ is a 3-layer MLP taking concatenated $\log_{\mathbf{0}}^c(\mathbf{x}_\tau)$, τ, sequence position t (t_i for training, t_p for inference), Ψ, and hyperbolic prefix $\mathbf{p}$ (from a causal GRU over target tangent states). Source cross-attention provides token-level alignment from the flow state to H_{enc}^{det}. H_{enc} is detached to insulate encoder features, while Ψ retains active gradient connections to f_ϕ to ground decoder conditioning in the continuous trajectory manifold.

B.4 Projective Quantiser

Two tied-weight quantisers operate on: (i) the 4-step-Euler flowed state $\hat{\mathbf{x}}_{flow}$; (ii) the initial source waypoint $\gamma_{\text{src}}(t_p)$ prior to any flow transport. This direct dual path is necessary to ensure that the foundational source geometry receives meaningful gradient signal early in training, even when the flow matching velocity field is still randomly initialized and hasn't yet converged to a semantic mapping. The ± 3 logit cap (Eqs. 5 and 6) is critical; without it, quantiser logits explode to $|\text{logits}| > 50$ and hijack decoding (Section F).

B.5 Inference

Algorithm 5 specifies decode-decoupled autoregressive inference. The geometric modules (flow matcher, quantiser) are computed during training only; they are *not* called at inference. Crucially, because the continuous geodesic trajectory and Ψ bottleneck are numerically integrated only once per source sentence at the encoder side, and the Riemannian flow matcher and quantiser are entirely decoupled during generation, per step autoregressive token decoding latency is virtually identical to that of a vanilla Transformer.

The decode-decoupled inference procedure is given in Algorithm 5 (Appendix C).

C Supplementary Algorithmic Procedures

This section details the complete algorithmic procedures for GeFLoSM. Specifically, Algorithm 1 details the Riemannian conditional flow matching (CFM) training step on the Poincaré ball; Algorithm 2 presents the projective quantisation procedure; Algorithm 3 provides the full end-to-end joint training

Algorithm 1 Riemannian CFM Training Step

Input: $\gamma_{\text{src}}(\cdot)^{det}$, $\gamma_{\text{tgt}}(\cdot)^{det}$, Ψ^{det}, prefix $\mathbf{p}$, velocity net v_θ, L_{tgt}
Output: $\mathcal{L}_{\text{CFM}}$

for $j = 1$ **to** L_{tgt} **do** ▷ $t_j = \frac{j-1}{L_{\text{tgt}}-1}$; sample $\gamma_{\text{src}}(t_j)$ via (Appendix A.2, Eq. 14)
  $\tau \sim \mathcal{U}[0, 0.95]$
  $\mathbf{x}_\tau \leftarrow \exp^c_{\gamma_{\text{src}}(t_j)}(\tau \cdot \log^c_{\gamma_{\text{src}}(t_j)}(\gamma_{\text{tgt}}(t_j)))$
  $\mathbf{u}_\tau \leftarrow \log^c_{\mathbf{x}_\tau}(\gamma_{\text{tgt}}(t_j))/\max(1-\tau, 0.1)$
  $\mathbf{v} \leftarrow v_\theta(\mathbf{x}_\tau, \tau, t_j, \Psi^{det}, \mathbf{p})$
  $\ell_j \leftarrow \log(1 + \|\mathbf{v} - \mathbf{u}_\tau\|^2)$
end for
return $\mathcal{L}_{\text{CFM}} = \frac{1}{L_{\text{tgt}}} \sum_j \ell_j$

Algorithm 2 Projective Quantisation (Training Only)

Input: $\gamma_{\text{src}}(\cdot)^{det}$, Ψ, v_θ, quantiser Q, target tokens $\mathbf{y}$
Output: $\mathcal{L}^{geom}_{CE}$

for $i = 1$ **to** L **do**
  $\hat{\mathbf{x}} \leftarrow \gamma_{\text{src}}(t_i)$ ▷ Initialize flowed state $\hat{\mathbf{x}}$ at source position t_i
  for $\tau = 0, \delta, 2\delta, 3\delta$ $(\delta = 1/4)$ **do**
    $\hat{\mathbf{x}} \leftarrow \exp^c_{\hat{\mathbf{x}}}(\delta \cdot v_\theta(\hat{\mathbf{x}}, \tau, t_i, \Psi, \mathbf{p}))$ ▷ Update integration state
  end for
  $\text{logits}_{flow} \leftarrow Q(\hat{\mathbf{x}})$ ▷ Apply projective quantiser Q (Eq. 5)
  $\text{logits}_{direct} \leftarrow Q(\gamma_{\text{src}}(t_i))$ ▷ Apply Q directly (Eq. 6)
  $\ell_i \leftarrow \text{CE}(\text{logits}_{flow}, y_i) + \text{CE}(\text{logits}_{direct}, y_i)$
end for
return $\mathcal{L}^{geom}_{CE} = \frac{1}{L} \sum_i \ell_i$

Algorithm 3 Full Joint Training Step

Input: Pair $(\mathbf{x}, \mathbf{y})$; networks $\text{Enc}, \text{Dec}_\Psi, f_\phi, v_\theta, Q$; weights $\lambda_{cfm}, \lambda_{vel}, \lambda_{geom}$
▷ Q: *projective quantiser* with parameters θ_Q (Phase C); maps $\mathbf{x} \in \mathbb{B}^d_c$ to vocabulary logits via $\log^c_{\mathbf{0}}$ projection, MLP(θ_Q), LayerNorm, and the ± 3 tanh cap (Eqs. 5–6)
Output: Updated θ

$H_{enc} \leftarrow \text{Enc}(\mathbf{x})$
$\gamma_{\text{src}}, \Psi, \mathcal{L}_{vel} \leftarrow$ Algorithm 4(H_{enc})
$\hat{\mathbf{y}} \leftarrow \text{Dec}_\Psi(\mathbf{y}_{<t_i}, H_{enc}, \Psi)$
$\mathcal{L}_{CE} \leftarrow \text{LabelSmooth}(\hat{\mathbf{y}}, \mathbf{y})$
$\tilde{H}_{enc} \leftarrow$ `stop_grad`(H_{enc})
$\tilde{\gamma}_{src} \leftarrow$ GeodesicSpline(Lift$(\tilde{H}_{enc})$)
▷ Ψ carries live gradients into f_ϕ and encoder
$\gamma_{\text{tgt}} \leftarrow$ GeodesicLift$(\text{Enc}(\mathbf{y})|_{\tilde{H}_{enc}})$ ▷ Lift targets via (Appendix A.2, Eqs. 13–14)
$\mathcal{L}_{\text{CFM}} \leftarrow$ Algorithm 1$(\tilde{\gamma}_{src}, \gamma_{\text{tgt}}, \Psi)$
$\mathcal{L}^{geom}_{CE} \leftarrow$ Algorithm 2$(\tilde{\gamma}_{src}, \Psi, v_\theta, Q, \mathbf{y})$
$\mathcal{L} \leftarrow \mathcal{L}_{CE} + \lambda_{cfm}\mathcal{L}_{\text{CFM}} + \lambda_{vel}\mathcal{L}_{vel} + \lambda_{geom}\mathcal{L}^{geom}_{CE}$
$\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}$; **return** θ

loop with encoder gradient detachment; Algorithm 4 defines the geodesic trajectory parameterisation and length-invariant bottleneck (Ψ) integration; and Algorithm 5 specifies the decode-decoupled autoregressive inference procedure.

Algorithm 4 Geodesic Trajectory and Invariant Bottleneck

Input: Encoder states H_{enc}, curvature c, ODE network f_ϕ
Output: Trajectory $\gamma_{\text{src}}(\cdot)$, invariant Ψ, velocity loss $\mathcal{L}_{vel}$

for $i = 1$ **to** L **do**
  $\mathbf{z}_i \leftarrow \exp_{\mathbf{0}}^{c}\big(\tanh(\|\mathbf{h}_i\|)\mathbf{h}_i/(\|\mathbf{h}_i\|\sqrt{d})\big)$; $t_i \leftarrow (i-1)/(L-1)$
end for
$\gamma_{\text{src}} \leftarrow$ GEODESICSPLINE$(\{\mathbf{z}_i\}, \{t_i\})$ ▷ Construct spline via (Appendix A.2, Eq. 14)
$\mathbf{h}(0) \leftarrow \mathbf{0}$; $\Delta\tau \leftarrow 1/L$
for $\tau = 0, \Delta\tau, \ldots, 1 - \Delta\tau$ **do**
  $\mathbf{h}(\tau{+}\Delta\tau) \leftarrow \mathbf{h}(\tau) + \Delta\tau \cdot f_\phi(\mathbf{h}(\tau), \gamma_{\text{src}}(\tau), \dot{\gamma_{\text{src}}}(\tau))$
end for
return $\gamma_{\text{src}}(\cdot)$, $\Psi = \mathbf{h}(1)$, $\mathcal{L}_{vel} = \sum_i \|\dot{\gamma_{\text{src}}}(t_i)\|_g^2$

D Experimental Setup

D.1 Datasets and Baselines

We evaluate on five benchmarks across four language pairs. **AMR-Transformer setting** (Li & Flanigan, 2022): WMT16 EN-DE (NC-v11 ≈238K and full ≈4.5M) and IWSLT15 EN-VI (≈133K, two test sets). **FlowSeq setting** (Ma et al., 2019): WMT14 EN-DE (≈4.5M), WMT16 EN-RO (≈610K), IWSLT14 DE-EN (≈150K), raw data only, no knowledge distillation.

Baselines in the discrete sequence translation setting include standard autoregressive Transformer (ours and parameter-doubled), deep architectures (Deep Transformer 16L/25L DLCL (Wang et al., 2019)), and graph-augmented semantic translation (Dual2seq (Song et al., 2019), Bi-LSTM+AMR (Nguyen et al., 2021), AMR-Transformer (Li & Flanigan, 2022)). For contemporary context, we also provide reference comparisons against large pretrained multilingual and LLM translators: mBART-50 (Liu et al., 2020), NLLB-200 (NLLB Team et al., 2022), LLaMA-2-7B, and ALMA-7B (Xu et al., 2024). In the continuous and generative sequence setting, we benchmark against iterative non-autoregressive CMLM-base (Ghazvininejad et al., 2019), LV-NAR (Shu et al., 2020), planar continuous normalizing flows FlowSeq-base and -large (Ma et al., 2019), and continuous diffusion sequence models (Li et al., 2022; Gao et al., 2024; Ye et al., 2024; Nguyen-Cong et al., 2026).

Following standard community benchmarking practice (Li & Flanigan, 2022; Ma et al., 2019), published baseline numbers (Dual2seq, Bi-LSTM+AMR, AMR-Transformer, mBART, and NLLB) are cited directly from their original authoritative publications under canonical test splits. To guarantee an unconfounded, strictly controlled evaluation free from pipeline or tokenization discrepancies, we also evaluate internally trained autoregressive Transformer baselines (standard and parameter-doubled) trained completely from scratch and scored under the exact same SentencePiece/BPE preprocessing, optimization schedule, and standardized SacreBLEU/chrF++/COMET evaluation protocol as GeFLoSM.

D.2 Implementation Details

We implement the core architecture using a 6 layer Transformer backbone with hidden dimension $d = 512$, 8 self attention heads, and an inner feed forward dimension of 2048. Models are trained using the Adam optimiser with a Noam learning rate schedule and a 10k step linear warmup

Algorithm 5 Decode-Decoupled Autoregressive Inference

Input: Source $\mathbf{x}$, trained $\{\text{Enc}, \text{Dec}_\Psi\}$, T_{max}
Output: Translation $\hat{\mathbf{y}}$

$H_{enc} \leftarrow \text{Enc}(\mathbf{x})$; $\gamma_{\text{src}}(\cdot), \Psi \leftarrow$ ALGORITHM 4(H_{enc})
$\hat{y}_0 \leftarrow \langle\text{BOS}\rangle$; $p \leftarrow 1$
while $\hat{y}_{p-1} \neq \langle\text{EOS}\rangle$ **and** $p \leq T_{max}$ **do**
  $\text{logits}_p \leftarrow \text{Dec}_\Psi(\hat{\mathbf{y}}_{<p}, H_{enc}, \Psi)$ ▷ transformer + Ψ-gate *only*
  $\hat{y}_p \leftarrow \arg\max_k(\text{logits}_p)_k$; $p \leftarrow p + 1$
end while
return $(\hat{y}_1, \ldots, \hat{y}_{p-1})$

period. For the Riemannian manifold, we set the Poincaré ball curvature to $c = 1.0$ and apply standard label smoothing with $\varepsilon = 0.1$. The composite objective balances loss components via fixed weighting coefficients: $\lambda_{cfm} = 0.02$ for flow matching, $\lambda_{vel} = 0.01$ for velocity regularisation, and $\lambda_{geom} = 0.1$ for auxiliary projective quantisation. During CFM training, flow time is sampled uniformly as $\tau \sim \mathcal{U}[0, 0.95]$, with numerical stability maintained via a denominator floor of 0.1 and $\log 1\mathrm{p}$ loss smoothing. Model checkpoints are evaluated at the conclusion of each training epoch, with the final model selected based on peak validation BLEU.

E Extended Results

E.1 Main Results — WMT16 English–German

Tables 1 and 2 evaluate GeFLoSM against sequence models, deep Transformers, AMR-augmented systems, and pretrained foundations on WMT16 EN–DE and low-resource IWSLT15 EN–VI across SacreBLEU, chrF++, and COMET. On WMT16 EN–DE, GeFLoSM achieves 23.4 / 49.1 / 78.1 on NC-v11 and 27.2 / 54.2 / 80.2 on full, establishing new state-of-the-art among from-scratch models and outperforming AMR-Transformer (+1.3 on NC-v11, +0.7 on full). On IWSLT15 EN–VI, GeFLoSM reaches 31.4 / 57.3 / 88.0 on `tst2013` and 29.1 / 51.8 / 78.2 on `tst2015`, surpassing AMR-Transformer (+0.8 / +0.9) across all three metrics. Notably, 6-layer GeFLoSM trained from scratch outperforms zero-shot LLaMA-2-7B (27.2 vs 19.0 SacreBLEU) and significantly narrows the margin to 50-layer pretrained multilingual foundations (mBART-50, NLLB-200).

E.2 Main Results — WMT14/16 and IWSLT14 (Raw Data)

Table 3 benchmarks sequence generation across distinct modeling paradigms on five standard translation datasets, organized into two explicit evaluation panels: **Part A** reinstates the complete legacy evaluation from Version 6 of this paper reporting the legacy metric tuple (BLEU ↑ / TER ↓ / Meteor ↑), while **Part B** provides the standardized modern evaluation reporting (SacreBLEU ↑ / chrF++ ↑ / COMET ↑). Each column represents a benchmark dataset, and each cell reports the corresponding 3-metric tuple.

We emphasize a fundamental distinction across generative sequence modeling paradigms: FlowSeq (Ma et al., 2019), CMLM (Ghazvininejad et al., 2019), LV-NAR (Shu et al., 2020), DiffusionLM (Li et al., 2022), Diffomer (Gao et al., 2024), DINOISER (Ye et al., 2024), and FastDiSS (Nguyen-Cong et al., 2026) explore non-autoregressive or continuous diffusion formulations. Crucially, while iterative diffusion models (DiffusionLM, Diffomer, DINOISER, FastDiSS) achieve higher raw SacreBLEU on select benchmarks, they require 5–20 iterative denoising steps per sentence at inference. In contrast, GeFLoSM incorporates Riemannian CFM on the Poincaré ball purely as a continuous training-time geometric regularizer to align sequence manifolds, decoding autoregressively at standard Transformer latency, with the geometric modules fully decoupled from the decode path ($O(1)$ additional cost; Algorithm 5, Appendix C). Under the compute-matched raw data training protocol evaluated by FlowSeq, GeFLoSM achieves uniform improvements (+1.05 BLEU on WMT14 EN–DE, +0.90 on DE–EN, +0.84 on WMT16 EN–RO, +1.11 on RO–EN over FlowSeq-large), confirming that continuous geodesic transport provides superior manifold representations compared to discrete planar normalizing flows without incurring multi-step sampling overhead. We report BLEU, TER, and Meteor scores for our model; since the baseline papers only report BLEU, unreported metrics for the baselines are marked with hyphens.

E.3 Complete Result Matrices

Tables 5–7 report the complete, untrimmed result matrices across all evaluated splits and benchmarks, comparing GeFLoSM against supervised sequence models, deep Transformers, pretrained multilingual foundations, and continuous generative modeling paradigms across both legacy metrics (BLEU, TER, Meteor) and standardized modern metrics (SacreBLEU, chrF++, COMET).

F Analysis: The Decode-Decoupling Requirement

Train/Inference Mismatch Mechanism.

Table 5: Test performance on WMT16 EN–DE across NC-v11 and Full splits, reporting both legacy literature metrics (BLEU ↑ / TER ↓ / Meteor ↑) and standardized modern multi-metric evaluation (SacreBLEU ↑ / chrF++ ↑ / COMET ↑). chrF++ uses word order 2 (Popović, 2017); COMET uses `wmt22-comet-da` (Rei et al., 2022).

	NC-v11 Split (238k pairs)		**Full Split (4.5M pairs)**	
Architecture	**Legacy** BLEU / TER / Meteor	**Modern** SacreB / chrF / COMET	**Legacy** BLEU / TER / Meteor	**Modern** SacreB / chrF / COMET
Supervised / Discrete Sequence Models (Trained from Scratch)				
Dual2seq (Song et al., 2019)	19.2 / 63.1 / 38.4	19.2 / 44.8 / –	25.5 / 54.8 / 43.8	25.5 / 51.2 / –
Bi-LSTM+AMR (Nguyen et al., 2021)	19.0 / 66.4 / 37.5	19.0 / 43.5 / –	24.8 / 58.9 / 43.1	24.8 / 50.1 / –
AMR-Transformer-DI (Li & Flanigan, 2022)	21.5 / 62.7 / 40.4	21.5 / 47.9 / –	26.4 / 56.7 / 44.9	26.4 / 52.8 / –
AMR-Transformer (MHA) (Li & Flanigan, 2022)	22.1 / 62.0 / 41.1	22.1 / 48.6 / –	26.5 / 56.4 / 45.2	26.5 / 53.1 / –
Vanilla Transformer	20.3 / 66.3 / 39.4	20.3 / 46.1 / 70.8	26.0 / 58.5 / 45.2	26.0 / 52.2 / 78.5
Vanilla Transf. (× params)	20.9 / 62.1 / 40.3	20.9 / 47.0 / 71.5	26.2 / 57.6 / 45.2	26.2 / 52.6 / 78.8
Deep Transf. (16L/25L) (Wang et al., 2019)	–	22.4 / 49.1 / 73.2	–	28.4 / 55.1 / 81.4
GeFLoSM (ours)	**23.4 / 58.7 / 42.1**	**23.4 / 49.1 / 78.1**	**27.2 / 55.0 / 46.0**	**27.2 / 54.2 / 80.2**
Pretrained Multilingual & LLM Context (Trillions Tokens / Pretrained)				
mBART-50 (Liu et al., 2020)	–	22.8 / 49.8 / 73.9	–	30.5 / 57.2 / 84.1
LLaMA-2-7B (0-shot) (Xu et al., 2024)	–	–	–	19.0 / 46.2 / 76.4
ALMA-7B (finetuned) (Xu et al., 2024)	–	–	–	30.3 / 57.8 / 85.6
NLLB-200 (NLLB Team et al., 2022)	–	–	–	33.0 / 59.2 / 85.8

Table 6: Test performance on IWSLT15 EN–VI across test splits, reporting both legacy literature metrics (BLEU ↑ / TER ↓ / Meteor ↑) and standardized modern multi-metric evaluation (SacreBLEU ↑ / chrF++ ↑ / COMET ↑). chrF++ uses word order 2; COMET uses `wmt22-comet-da` (scaled ×100).

	tst2013 (Spoken Development / Test)		**tst2015 (Official Test Set)**	
Architecture	**Legacy** BLEU / TER / Meteor	**Modern** SacreB / chrF / COMET	**Legacy** BLEU / TER / Meteor	**Modern** SacreB / chrF / COMET
Supervised / Discrete Sequence Models (Trained from Scratch)				
Bi-LSTM+AMR (Nguyen et al., 2021)	26.4 / 56.4 / 44.1	26.4 / 47.8 / –	25.2 / 60.5 / 42.1	25.2 / 46.1 / –
Vanilla Transformer	30.0 / 52.1 / 48.2	30.0 / 52.1 / 78.4	27.6 / 57.6 / 45.4	27.6 / 49.8 / 75.8
Vanilla Transf. (× params)	28.3 / 54.4 / 46.4	28.3 / 50.4 / 76.9	26.8 / 59.2 / 44.2	26.8 / 48.7 / 74.5
AMR-Transformer-DI (Li & Flanigan, 2022)	30.2 / 52.4 / 48.2	30.2 / 52.4 / –	28.2 / 57.3 / 45.5	28.2 / 50.3 / –
AMR-Transformer (MHA) (Li & Flanigan, 2022)	30.6 / 52.1 / 48.5	30.6 / 52.8 / –	28.2 / 57.1 / 45.9	28.2 / 50.5 / –
GeFLoSM (ours)	**31.4 / 51.0 / 49.5**	**31.4 / 57.3 / 88.0**	**29.1 / 55.6 / 47.0**	**29.1 / 51.8 / 78.2**
Pretrained Multilingual Benchmarks (External Bitext / Monolingual)				
mBART-25 (Liu et al., 2020)	–	–	–	35.4 / 54.2 / 83.1
NLLB-200 (NLLB Team et al., 2022)	–	–	–	34.8 / 53.7 / 82.5

The projective quantiser trains against teacher-forced target geometry $\gamma_{\text{tgt}}(t_i)$ from gold target hidden states. At inference, the quantiser operates without gold targets on flowed states derived from source geometry, which are semantically misaligned early in training.

Near initialisation, quantiser logits are small ($\sigma \approx 0.03$, $|\text{logits}| \ll 1$) and the argmax is dominated by the transformer decoder (BLEU ≈ 0.59). Under teacher forcing, quantiser logits grow unbounded: without the ± 3 cap, our probe measured $|\text{logits}_{max}| = 51$ (vs. transformer $\sigma \approx 1.0$) after 120 steps, hijacking the argmax. With the cap but without decode decoupling, the quantiser still dominates at inference by epoch 3, causing BLEU$\to 0$ despite falling training loss.

Remark 1. *This failure mode exemplifies exposure bias / train-inference mismatch (Ranzato et al., 2016): auxiliary objectives conditioned on teacher-forced targets overfit to training-only signals if retained in the decode path; decode decoupling provides the structural remedy.*

G Supplementary Material and Code Access

To guarantee full reproducibility of all empirical findings and ensure strict compliance with double blind review policies, the complete source code for **GeFLoSM** is provided through two complemen-

Table 7: Benchmarking sequence generation paradigms across standard translation benchmarks. **Part A** reinstates the legacy evaluation from Version 6 (BLEU ↑ / TER ↓ / Meteor ↑). **Part B** provides standardized modern evaluation (SacreBLEU ↑ / chrF++ ↑ / COMET ↑). Baselines cited from (Ghazvininejad et al., 2019; Shu et al., 2020; Ma et al., 2019; Li et al., 2022; Gao et al., 2024; Ye et al., 2024; Nguyen-Cong et al., 2026). Baselines report SacreBLEU only (– for chrF++/COMET). Signature: `nrefs:1|case:mixed|eff:no|tok:13a|smooth:exp|version:2.4.0` (Post, 2018).

Architecture	Paradigm / Steps	WMT14 EN–DE	WMT14 DE–EN	WMT16 EN–RO	WMT16 RO–EN	IWSLT14 DE–EN
Part A: Legacy Literature Evaluation (Version 6 Reinstated) — Tuple: (BLEU ↑ / TER ↓ / Meteor ↑)						
CMLM-base (Ghazvininejad et al., 2019)	Iterative NAR (10 steps)	10.88 / – / –	– / – / –	20.24 / – / –	– / – / –	– / – / –
LV NAR (Shu et al., 2020)	Iterative NAR (4 steps)	11.80 / – / –	– / – / –	– / – / –	– / – / –	– / – / –
FlowSeq-base (Ma et al., 2019)	Continuous Flow (1 pass)	18.55 / – / –	23.36 / – / –	29.26 / – / –	30.16 / – / –	24.75 / – / –
FlowSeq-large (Ma et al., 2019)	Continuous Flow (1 pass)	20.85 / – / –	25.40 / – / –	29.86 / – / –	30.69 / – / –	– / – / –
GeFLoSM (ours, raw)	**AR (beam = 5)**	**21.9 / 61.2 / 41.5**	**26.3 / 55.4 / 45.2**	**30.7 / 50.8 / 49.3**	**31.8 / 49.2 / 50.6**	**26.1 / 56.1 / 45.8**
Part B: Standardized Modern Evaluation (Continuous Generative Baselines) — Tuple: (SacreBLEU ↑ / chrF++ ↑ / COMET ↑)						
CMLM-base (Ghazvininejad et al., 2019)	Iterative NAR (10 steps)	24.34 / – / –	28.71 / – / –	31.26 / – / –	31.13 / – / –	29.41 / – / –
LV-NAR (Shu et al., 2020)	Iterative NAR (4 steps)	24.20 / – / –	– / – / –	– / – / –	– / – / –	– / – / –
FlowSeq-base (Ma et al., 2019)	Continuous Flow (1 pass)	18.55 / – / –	23.36 / – / –	29.26 / – / –	30.16 / – / –	24.75 / – / –
FlowSeq-large (Ma et al., 2019)	Continuous Flow (1 pass)	20.85 / – / –	25.40 / – / –	29.86 / – / –	30.69 / – / –	– / – / –
DiffusionLM (Li et al., 2022)	Continuous Diff. (20 steps)	22.91 / – / –	19.69 / – / –	29.39 / – / –	30.17 / – / –	29.11 / – / –
Diffomer (Gao et al., 2024)	Latent Diff. (20 steps)	23.31 / – / –	25.30 / – / –	29.20 / – / –	29.37 / – / –	28.01 / – / –
DINOISER (Ye et al., 2024)	Noise-Manipulated (20 steps)	25.70 / – / –	29.05 / – / –	31.08 / – / –	31.22 / – / –	31.61 / – / –
FastDiSS (Nguyen-Cong et al., 2026)	Few-Step Diff. (5 steps)	26.02 / – / –	29.75 / – / –	30.90 / – / –	31.81 / – / –	32.70 / – / –
FastDiSS (Nguyen-Cong et al., 2026)	Few-Step Diff. (20 steps)	26.39 / – / –	29.83 / – / –	31.44 / – / –	31.99 / – / –	32.88 / – / –
GeFLoSM (ours, raw)	**CFM Reg. + AR (beam = 5)**	**21.90 / 48.5 / 74.8**	**26.30 / 52.1 / 79.1**	**30.70 / 54.8 / 82.4**	**31.80 / 55.9 / 83.5**	**26.10 / 51.5 / 82.0**

tary avenues. An anonymous, open access repository containing the complete, version controlled PyTorch implementation is publicly accessible at `https://anonymous.4open.science/r/HFM_SM_arch-E564/`, where reviewers can directly inspect all architectural modules and Riemannian flow matching operators.

In addition, for offline evaluation and archival self containment, the identical codebase has been packaged as a standalone archive (`HFM_SM_code.zip`) and submitted directly to the conference submission portal as supplementary material.