

Efficient One-to-Many Translation with Joint Multi-Stream Diffusion

Yiwen Guan, Jacob Whitehill

Worcester Polytechnic Institute

yguan2@wpi.edu, jrwhitehill@wpi.edu

Abstract

One-to-many machine translation (MT) is computationally expensive for autoregressive (AR) systems, which suffer from linear latency scaling with both sequence length and the number of target languages. We explore how diffusion can enable multilingual translation with a discrete diffusion framework that refines all target languages in parallel, achieving sublinear latency scaling with the number of targets, and supports deployment as a single unified model to replace multiple independent systems. Conditioned on a continuous semantic anchor rather than source tokens, our framework supports zero-shot transfer to unseen source languages without retraining, maintaining approximately 75% of its supervised translation quality on zero-shot sources. We investigate the quality-latency frontier and find that with accelerated sampling, it achieves comparable supervised quality to AR baselines with a $2\times$ speedup and 11.9% better zero-shot BLEU. These results highlight the potential of joint multi-stream diffusion as a practical and flexible alternative for efficient one-to-many translation.

1 Introduction

Real-world multilingual translation, such as in live interpreting for international meetings (e.g., the United Nations, the European Parliament), frequently requires different target languages simultaneously from a single source stream. In these scenarios, the set of requested languages can change dynamically as different client nodes join or leave the session, demanding a flexible and low-latency framework. However, existing approaches predominantly rely on Transformer-based (Vaswani et al., 2017) autoregressive (AR) architectures, typically either deploying separate one-to-one models for each language pair, or exploiting a unified model managing multilingualism via language-specific control tokens (John-

son et al., 2017; Fan et al., 2021). While they attain high accuracy, AR models generate tokens sequentially and require separate decoding passes for each target language. This computational bottleneck makes AR models expensive for parallel one-to-many translation, especially in resource-constrained applications, as inference latency scales linearly with both the sequence length and the number of targets (Gu et al., 2018; Kasai et al., 2020).

To mitigate this issue, non-autoregressive (NAR) translation has been recently investigated (Gu et al., 2018; Ghazvininejad et al., 2019; Xiao et al., 2023). By generating tokens in parallel, NAR methods offer compelling speed-quality trade-offs (Qian et al., 2021). However, most existing methods assume conditional independence among target tokens, resulting in the “multimodality problem” (Gu et al., 2018). Furthermore, naively adapting these frameworks to one-to-many translation (e.g., running multiple models in parallel) is both memory intensive and computationally inefficient.

As an alternative, diffusion models (DMs) have emerged as a powerful NAR paradigm for sequence generation through iterative parallel refinement (Ho et al., 2020). Continuous DMs map tokens into a continuous embedding space (Li et al., 2022; Gong et al., 2023), while discrete DMs operate directly on the categorical vocabulary (Austin et al., 2021; Nie et al., 2025). Despite their potential, applying DMs to machine translation (MT) remains underexplored. XDLM (Chen et al., 2023) uses discrete diffusion with cross-lingual pretraining but is limited to one-to-one tasks. Scaling such diffusion frameworks raises important design questions of the optimal input representation, attention mechanism, and diffusion inference schedule.

In this work, we explore the viability and design space of parallel multilingual MT. As an in-

stantiation, we propose **PrismDiff**, a discrete diffusion framework for parallel one-to-many generation. Metaphorically, PrismDiff acts like a prism: it “refracts” a shared source representation into multiple target outputs through parallel diffusion refinement. In multilingual MT tasks, PrismDiff harnesses a language-agnostic semantic anchor to guide the diffusion process, rather than conditioning directly on source tokens. This shared anchor not only enables zero-shot transfer capability to unseen source languages but also improves translation accuracy.

Our contributions are summarized as follows:

1. We propose PrismDiff, a multi-stream discrete diffusion framework for one-to-many translation, which refines multiple target languages in parallel from a shared semantic anchor. Joint optimization over all target streams provides implicit cross-lingual regularization, achieving better translation quality than running independent one-to-one MT systems.
2. We explore the quality-latency frontier of diffusion systems against AR baselines. We show that under supervised settings, PrismDiff achieves a competitive frontier and outperforms AR systems at matched low-latency settings. Furthermore, under source-side zero-shot settings, the framework forms a stronger performance frontier than AR baselines.
3. Through systematic ablations, we show that effective zero-shot transfer requires a high-quality cross-lingual semantic space rather than an arbitrary multilingual encoder, and the anchor mechanism is not replaceable by increasing the number of diffusion steps alone.

2 Related Work

2.1 Non-autoregressive Translation

Non-autoregressive translation (NAT) aims to reduce the sequential bottleneck of autoregressive decoding by predicting target tokens in parallel (Gu et al., 2018; Guo et al., 2019; Ghazvininejad et al., 2019). This set of methods has produced strong speed-quality trade-offs, especially through conditional masked language modeling and iterative refinement. Several strong NAT systems have

been proposed for one-to-one translation, for example, conditional masked language models repeatedly update masked positions conditioned on the source sentence (Ghazvininejad et al., 2019), while glancing training exposes the model to a curriculum of partially observed target tokens (Qian et al., 2021). Although these methods establish compelling quality-latency trade-offs in bilingual settings, they are not directly designed for parallel one-to-many generation, and do not provide a natural mechanism for sharing a single semantic representation across multiple target streams. Similarly, XDLM (Chen et al., 2023) explores discrete diffusion for cross-lingual generation focusing on one-to-one translation. Our work targets different settings: a unified model decoding into multiple target languages in parallel, which motivates different design choices and baselines. Previous systems for multilingual translation rely on task prompts or specialized architectures to coordinate target languages (Johnson et al., 2017; Fan et al., 2021; Azpiazu and Pera, 2020; Guan and Whitehill, 2025), which are limited in deployment flexibility. In contrast, our work uses a single diffusion canvas in which multiple target streams are optimized together, allowing for joint one-to-many decoding.

2.2 Diffusion Language Model

In the context of language modeling, diffusion language models (DLMs) generally fall into two categories: continuous and discrete. Continuous DLMs operate in embedding space and learn to denoise latent token representations (Li et al., 2022; Gong et al., 2023). Discrete DLMs instead define the corruption and reverse processes directly over categorical tokens, often using masking or transition matrices (Austin et al., 2021; He et al., 2023; Nie et al., 2025; Bie et al., 2025; Ye et al., 2025). Our method follows the discrete masked diffusion paradigm, which naturally supports parallel token reconstruction and variable sampling budgets.

Diffusion-based translation remains less explored than AR and other NAT methods. XDLM (Chen et al., 2023) studies cross-lingual discrete diffusion with pretraining, but focuses primarily on one-to-one generation. In contrast, we target parallel one-to-many translation, where the model must generate multiple target languages for the same source sentence concurrently; we also emphasize the quality-latency frontier exposed by changing the number of sampling steps on the

same model.

2.3 Inference-time Acceleration for Diffusion Models

A practical limitation of DMs is that generation quality depends on the number of denoising steps, creating a trade-off between latency and output quality. Prior work has explored techniques to accelerate discrete diffusion inference speed while maintaining quality (Chen et al., 2025; Liu et al., 2025; Wu et al., 2025; Bartosh et al., 2026), including training-free acceleration strategies, such as non-uniform schedules and jumpy sampling (Yeh et al., 2024; Chen et al., 2024; Irwin et al., 2025).

In our work, we adopt the logarithm-uniform schedule applied in Irwin et al. 2025 as a control knob at inference time, allowing a single trained model to operate at multiple quality-latency points by varying the number of sampling steps, without modifying model capacity or retraining the model. This is the key mechanism we use to compare with AR baselines, where latency reduction requires shrinking model depth.

3 Methodology

3.1 Preliminaries

We adopt the framework of existing discrete diffusion models, which generate outputs through iterative denoising of masked sequences (Austin et al., 2021; Nie et al., 2025; Bie et al., 2025). However, we focus on one-to-many generation, conditioned on a sequence of latent representations instead of source tokens.

Given a source sequence X with length L_{src} , our goal is to generate K target translations $\mathcal{Y} = \{Y^{(1)}, Y^{(2)}, \dots, Y^{(K)}\}$ in parallel, where the k -th target language sequence $Y^{(k)} = (y_1^k, \dots, y_{L_k}^k)$ consists of discrete tokens from a vocabulary $\mathcal{V}$. Following existing discrete DLMs, we represent each example as a concatenate a prompt p (conditioned information) and a response r (target to be generated). In our setting, p remains unmasked, while r is progressively corrupted and reconstructed.

The forward process is a discrete masked corruption process defined over T discrete timesteps $t \in \{1, \dots, T\}$, where T controls the total amount of noise applied during training. Let r_0 be the clean response and r_t be the corrupted response at timestep t , where $r_{t,i}$ denotes the token at the position index i in the generated sequence at dif-

fusion timestep t . The forward process independently applies [MASK]s to the clean response r_0 to produce the corrupted state r_t , based on a schedule $\gamma(t)$ (i.e., the probability of each token being masked at timestep t), where $\gamma(T) = 1$ results in fully masked sequence.

The reverse process employs a neural network $p_\theta(\cdot)$ to denoise the sequence, which takes the concatenation $[p; r_t]$ of the prompt p and the corrupted sequence r_t , and predicts the probability of the original response token $p_\theta(r_{0,i} | p, r_t)$ for each masked position $i \in \mathcal{M}_t$, where $\mathcal{M}_t$ denotes all masked positions at time t . The training objective is to maximize the variational lower bound of data log-likelihood, optimized via a cross-entropy loss restricted to masked tokens:

$$\mathcal{L}(\theta) \triangleq -\mathbb{E}_{t,p,r_0,r_t} \left[\sum_{i \in \mathcal{M}_t} \log p_\theta(r_{0,i} | p, r_t) \right] \quad (1)$$

The inference stage begins from a fully masked sequence, and iteratively denoises it over multiple refinement steps. At each step, the model first predicts the clean sequence from the current noisy state, then performs a remasking process that selects a fraction of tokens to be masked again, ensuring the sequence becomes progressively less noisy.

3.2 Proposed Method

An illustration of our framework is in Figure 1.

Language-Agnostic Semantic Anchor. Conditioning on source tokens would bind the model to a specific source language, preventing zero-shot transfer. To decouple the generation process from source language surface forms and enable zero-shot transfer, we condition the diffusion process on a continuous representation rather than on source tokens directly. Specifically, we extract the last hidden states $H \in \mathbb{R}^{L_{\text{src}} \times d_{\text{model}}}$ from the source input X using LaBSE (Language-agnostic BERT Sentence Embedding) (Feng et al., 2022), a pretrained multilingual encoder optimized to align parallel sentences across languages into a shared embedding space. Then H is projected into the diffusion model’s dimension d_{model} via a learnable linear projection to obtain the anchor $Z \in \mathbb{R}^{L_{\text{src}} \times d_{\text{model}}}$.

Multi-Stream Generation. To enable parallel generation, we construct a unified “canvas” C_t at timestep $t \in [0, T]$. This canvas concatenates the

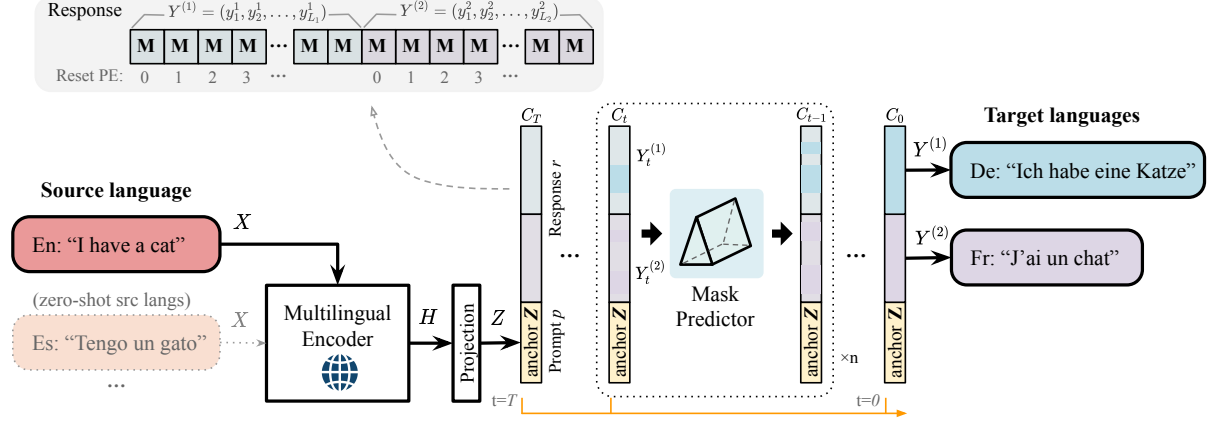


Figure 1: **An illustration of PrismDiff.** “Reset PE” is short for Reset Positional Encoding. PrismDiff takes one source language sentence X as input, generating multiple target languages $\{Y^{(1)}, Y^{(2)}, \dots, Y^{(K)}\}$ concurrently. The source language can be replaced with other zero-shot languages, such as Spanish. For simplicity, we use En-De, Fr} translation as an example. In practice, the total diffusion refinement steps can be reduced to accelerate.

anchor Z with all K noisy target streams: $C_t = [p_0; r_t] = [Z; Y_t^{(1)}; \dots; Y_t^{(K)}]$ where $Y_t^{(k)}$ represents the k -th target language stream at diffusion step t . The neural network processes the entire C_t simultaneously and outputs predictions for all masked positions across all streams. Masked tokens in the k -th stream are replaced by a language-specific mask token $[\text{MASK}]_k$, added as special tokens to the shared BPE vocabulary, rather than a generic mask.

Full cross-stream attention introduces interference between target languages during generation. We therefore replace full cross-stream attention with exclusive attention within each stream. While the attention is isolated, all streams share the same model parameters and are jointly optimized, providing implicit cross-lingual regularization that benefits each individual stream.

Since the canvas concatenates multiple sequences, standard positional encoding would assign monotonically increasing indices. Instead, we employ Reset Positional Encoding (E_{pos}), as illustrated in the upper left corner of Figure 1. For the anchor Z , position indices range from 0 to $L_{\text{src}} - 1$. For each target stream $Y^{(k)}$, we restart the position indices from 0 to $L_{\text{tgt}} - 1$. This design allows for the dynamic reordering, removal or addition of target languages without affecting performance, and encourages the model to correctly identify token positions within individual streams.

Training. During training, we sample a timestep $t \sim \mathcal{U}(1, T)$ and corrupt the target streams, while the anchor Z remains uncorrupted. The model predicts the original tokens from all masked positions

across all streams in parallel. The loss function is adapted from Eq. 1:

$$\mathcal{L}(\theta) = -\mathbb{E}_{t, Z, r_0, r_t} \left[\sum_{i \in \mathcal{M}_t} \log p_{\theta}(r_{0,i} | Z, r_t) \right] \quad (2)$$

where $r_t = [Y_t^{(1)}; \dots; Y_t^{(K)}]$ is the concatenation of all target streams.

Inference. In the inference phase, we start with a fully masked canvas C_T with length L containing only the anchor Z and language-specific masks. At each refinement step, the model retains the top $(1 - m) \times L$ most confident tokens and re-masks the remainder, where the masking ratio m is determined by the schedule. We experiment on both the vanilla schedule and a logarithm-uniform schedule following Irwin et al. 2025 to progressively unmask tokens. Specifically, the logarithm-uniform mask distribution over the masking ratio m is defined as:

$$\log m \sim \mathcal{U} \left(\log \frac{1}{L}, 0 \right) \quad (3)$$

, the masking ratio m is bounded within $[1/L, 1]$, and $1/L$ corresponds to the single-token mask limit. See more details in A.1.

Since discrete diffusion models lack explicit autoregressive constraints, they may generate consecutive repetitive tokens. To mitigate this, we apply Connectionist Temporal Classification (CTC) decoding (Graves et al., 2006) as a post-processing step without training on CTC loss. Specifically, we use the CTC collapse rule (removing consecutive duplicates and blanks) on the final generated

sequence, which effectively acts as a structural regularization for NAR output.

4 Experimental Setup

4.1 Baselines

We compare PrismDiff with the following AR and diffusion baselines:

- **AR**: Decoder-only AR Transformers conditioned on the same source-side representations (anchor Z), and trained on all En-X pairs with task prompts. We vary the number of decoder layers to trace the AR quality-latency frontier. Our preliminary experiments (detailed in A.3) show that this architecture outperforms other AR settings such as encoder-decoder.
- **Diff-Indep**: Independent diffusion decoders conditioned on the same anchor Z , but trained separately for each En-X pair. This isolates the contribution of joint multi-stream optimization from the anchor mechanism.
- **Diff-SrcTokens**: A multi-stream diffusion model identical to PrismDiff, but conditioned on source tokens instead of anchor Z . This tests whether the gains of PrismDiff come from anchor-based semantic conditioning.
- **Transformer-Encoder Trees (TET)** : A strong concurrent one-to-many NAT model (Guan and Whitehill, 2025) that generates multiple targets with a unified encoder-tree architecture with CTC. TET is trained on En-X tasks but lacks zero-shot transfer capability.

We do not directly compare against one-to-one NAT systems such as GLAT and XDLM in a one-to-many setting, as our focus is on isolating specific design decisions within a unified diffusion framework. The Diff-Indep is designed for controllable comparisons: it uses the same architecture and anchor as PrismDiff but generates each languages independently.

4.2 Datasets and Evaluation Metrics

Datasets. We utilize Multi30K (Elliott et al., 2016) as our primary benchmark. Multi30K is a multilingual dataset of image descriptions with short, visually grounded sentences. We use the English-to-German and English-to-French (En-{De, Fr}) pairs (29k) for training, and 1k

for testing. To evaluate zero-shot transfer performance, we create a synthetic Spanish-to-X (Es-X) test set by translating English into Spanish using NLLB-200 (Costa-Jussà et al., 2022).

To further verify the effectiveness of anchor mechanism, we conduct supplementary experiments on Europarl (Koehn, 2005), a large-scale, aligned parallel corpus from European Parliament proceedings. We construct a training set of 260k English-to-French, Dutch, Romanian, and Danish examples (En-{Fr, Nl, Ro, Da}), and 33k for testing, with Spanish serving as the unseen source language. Europarl’s Spanish evaluation set comes from original human-authored corpus, allowing us to rigorously verify our zero-shot transfer claims.

Metrics. Translation quality and cross-lingual semantic transfer are measured with SacreBLEU¹ (Post, 2018) and COMET² (Rei et al., 2020). Inference efficiency is reported as the average end-to-end wall-clock latency (ms) for generating all requested target sentences for a source sentence on a single model instance. Note that AR systems can reduce wall-clock latency by running multiple model replicas in parallel, which increases memory and deployment complexity. Our comparison focuses on one-to-many latency in a single model.

4.3 Implementation Details

All diffusion systems use an 8-layer Transformer encoder backbone ($d_{model} = 768$, $n_{head} = 8$) to serve as the diffusion decoder; each contains about 230M parameters. Unless otherwise noted, the diffusion systems use a total number of diffusion steps $T = 100$ on Multi30K and $T = 40$ on Europarl with accelerated sampling at different numbers of sampling steps. This allows a single trained model to exhibit multiple operating points by adjusting the number of sampling steps t . We also report $T = 10$ diffusion results as supporting comparisons. We set the max length of diffusion blocks $L_{src} = 40$ for Multi30K and $L_{src} = 80$ for Europarl.

The decoder-only AR models use similar Transformer settings, contain about 247M, 216M, and 208M parameters in the 8-layer, 4-layer, 3-layer versions, respectively. AR models use the same max length as diffusion systems.

The Transformer-Encoder Tree (TET) model

¹The SacreBLEU signature is nrefs:1|case:mixed|eff:no|tok:13a|smooth:exp|version:2.5.1.

²COMET is with Unbabel/wmt22-comet-da (v2.2.7)

(Guan and Whitehill, 2025) for Europarl has 4 leaf nodes (corresponding to the targets), each with a depth of 6. It contains 16 distinct Transformer encoder layers in total, distributed across its tree topology, resulting in a model size of 556M.

LaBSE is kept frozen during training; only the linear projection layer is trained as an adapter. The models are optimized using AdamW (Loshchilov and Hutter, 2019) with a learning rate of 5e-5. Models are trained for 60 epochs, and models with the best validation loss are used for evaluation. All models are trained and evaluated on a single NVIDIA L40S GPU, using batch size 8 during training and batch size 1 for evaluation.

Decoding Protocol. All models use a BPE vocabulary across all target languages. In the inference phase of all diffusion models, CTC collapse (as mentioned in Sec. 3.2) is applied after tokenization decoding. We find this technique consistently outperforms (by an average of 2.5%) vanilla decoding methods by removing consecutive duplicate tokens at the subword level. See A.3 for detailed results. For both diffusion and AR systems, we truncate each decoded sequence at the first EOS token.

5 Results and Analysis

Table 1 and Table 2 summarize the performance on Multi30K and Europarl. Rather than comparing a single decoding configuration, we analyze the quality–latency frontier induced by diffusion sampling steps and AR decoder depth. Compared with strong AR baselines, PrismDiff achieves a competitive quality–latency trade-off: it reaches comparable SacreBLEU and COMET scores with substantially lower latency, while showing advantages in source-side zero-shot transfer and parallel one-to-many generation.

5.1 Supervised Quality-Latency Frontier

Figure 2 (left) compares PrismDiff with strong decoder-only AR baselines. On supervised En-X tasks, PrismDiff reaches most of its BLEU gains within 25–30 sampling steps (t), corresponding to roughly 70–100 ms latency. Beyond this regime, additional sampling steps only bring marginal BLEU gains, while PrismDiff remains competitive with the best AR operating points over a wide latency range.

The key distinction between the two frontiers is the underlying trade-off mechanisms. To achieve

Src.	Model	t	BLEU	COMET	Latency
En	Diff-Indep	50	35.95	0.710	128.8
		30	35.55	0.710	76.6
		25	35.60	0.705	63.5
	Diff-SrcTokens	50	35.03	0.695	122.5
		30	35.07	0.690	72.6
		25	34.40	0.685	60.2
	AR (8-layer)		37.25	0.720	148.3
	AR (4-layer)		36.03	0.720	92.5
	AR (3-layer)		35.48	0.710	76.0
	PrismDiff	50	37.06	0.720	147.2
		30	37.09	0.715	83.2
		25	36.97	0.715	68.7
Es	Diff-Indep	50	27.21	0.670	128.5
		30	26.88	0.670	76.5
		25	26.94	0.670	63.7
	Diff-SrcTokens	50	0.14	0.305	122.8
		30	0.12	0.300	72.5
		25	0.25	0.300	62.5
	AR (8-layer)		25.22	0.655	154.5
	AR (4-layer)		26.03	0.670	92.5
	AR (3-layer)		24.54	0.655	77.6
	PrismDiff	50	28.23	0.674	141.7
		30	28.30	0.670	83.1
		25	28.22	0.670	69.3

Table 1: **Main results on Multi30K.** Metrics are averaged on De and Fr. Es is a zero-shot source language. t is the accelerated sampling steps using logarithm-uniform scheduling. All diffusion systems are trained with $T = 100$. Latency is reported in ms. Bold text denotes the best results under comparable latency budgets (or equivalent sampling steps t). See A.3 for details.

faster inference, AR models must reduce decoder depth (model capacity), leading to a substantial drop in BLEU. In contrast, diffusion-based systems obtain faster operating points by reducing the number of refinement steps at inference time, without sacrificing model capacity. The left portion of the curve highlights this advantage: under strict low-latency budgets, PrismDiff gracefully maintains near-peak translation quality (~ 37 BLEU), whereas the AR frontier degrades sharply. Hence, this comparison does not claim that diffusion is superior to AR in every supervised setting; rather, it shows that diffusion systems provide a competitive performance frontier and a flexible control knob.

5.2 Source-Side Zero-Shot Transfer

The architectural advantages become even more pronounced under source-side zero-shot trans-

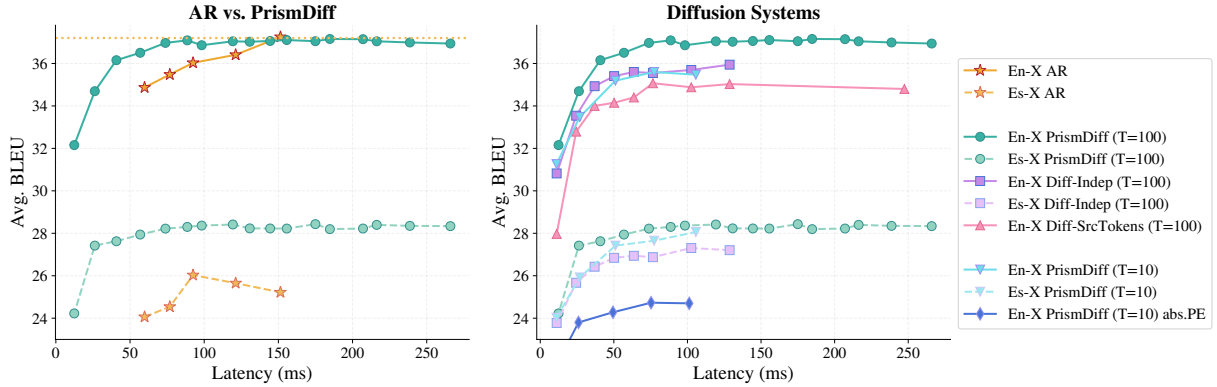


Figure 2: **Quality-latency frontiers on Multi30K.** Solid curves denote supervised En-X results, and dashed curves denote source-side zero-shot Es-X results. **Left:** Comparison between AR models and PrismDiff. **Right:** Comparison among diffusion-based systems. We omit the Es-X results for Diff-SrcTokens as its BLEU is < 1 . T indicates the total diffusion steps, and the latency operating points are obtained by varying the number of sampling steps. For AR models, operating points are obtained by varying the decoder depth from 3 to 8.

Src.	Model	t	BLEU	COMET	Latency
En	TET		26.64	0.630	12.1
	Diff-SrcTokens	40	26.23	0.728	224.8
		20	24.48	0.735	93.7
	PrismDiff	40	29.54	0.778	239.3
		20	28.84	0.785	112.7
Es	TET		0.14	0.265	11.4
	Diff-SrcTokens	40	0.03	0.320	225.6
		20	0.33	0.320	95.8
	PrismDiff	40	23.47	0.745	237.0
		20	22.10	0.745	112.1

Table 2: **Main results on Europarl.** Metrics are averaged on Fr, Nl, Ro, and Da. Es is a zero-shot source language. All diffusion systems are trained with $T = 40$. TET generates all targets in a single CTC pass without sequential or iterative steps, thus achieving low latency.

fer. As shown in Figure 2, when evaluating the English-trained systems on unseen Spanish sources (Es-X), the PrismDiff frontier consistently outperforms the AR baselines across all points. Because both models utilize the identical source-side semantic representation, this performance gap cannot be attributed to differences in input features. Instead, it suggests that iterative refinement over a shared semantic anchor is more robust to source-side domain shift than traditional left-to-right decoding.

We speculate that this is due to the cumulative effect of generation errors. In AR models, out-of-distribution inputs may trigger early decoding errors, which can cascade due to autoregression. In contrast, PrismDiff’s bidirectional context and iterative refinement mitigate such error accumulation, resulting in a more stable generation process.

This robustness is important for real-world deployment, where models trained on high-resource source languages encounter input from unseen sources. While PrismDiff does not completely bridge the supervised-to-zero-shot gap, it shows a stronger zero-shot frontier, suggesting that iterative refinement can more effectively exploit the semantic meaning regardless of the source language.

5.3 Ablation Studies

Effect of Semantic Anchor. Figure 2 and Table 1 isolate the effect of the anchor from increasing T . A Diff-SrcTokens model trained with $T = 10$ peaks at 33.84 BLEU; increasing to $T = 100$ improves this to 35.07. Under the same $T = 100$ setting, PrismDiff consistently outperforms Diff-SrcTokens across matched latency budgets in supervised scenarios, confirming that a larger T improves the sampling frontier but cannot fully explain PrismDiff’s gains. The most important divergence emerges in zero-shot scenarios: without the anchor, Diff-SrcTokens degrades severely ($\text{BLEU} < 1$ on Es-X), whereas PrismDiff maintains strong performance. Thus, a larger T and the semantic anchor play complementary roles: the former improves latency-quality trade-offs, whereas the latter provides the necessary cross-lingual mapping for zero-shot transfer.

Interestingly, PrismDiff performs better with exclusive attention than full attention, whereas Diff-SrcTokens shows the opposite trend. This suggests that without the anchor, cross-stream attention serves as a necessary channel for implicit cross-lingual alignment; once the anchor fulfills this alignment, cross-stream attention becomes re-

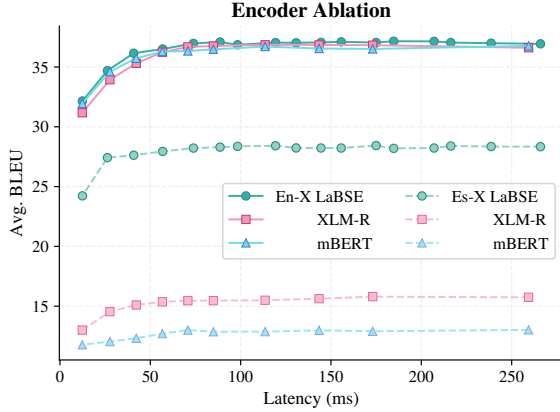


Figure 3: **Encoder ablation** for PrismDiff ($T = 100$) on Multi30K. Solid curves denote supervised En-X frontiers, and dashed curves denote zero-shot Es-X frontiers.

dundant and may introduce interference between language-specific surface forms. (See Table 5)

Effect of Joint Multi-Stream Optimization.

While Diff-Indep uses the same diffusion mechanism and semantic anchor under matched $T = 100$ setting, it generates each target language independently. This ablation therefore clarifies whether PrismDiff’s gains come solely from the anchored diffusion process. As shown in Figure 2 and Table 1, PrismDiff consistently outperforms Diff-Indep in both supervised and zero-shot settings: at $t = 30$, it maintains a ~ 1.5 and ~ 1.4 BLEU advantage over Diff-Indep on supervised and zero-shot setting, respectively. This suggests that jointly optimizing all streams provides not only deployment convenience for one-to-many translation, but also implicit cross-lingual regularization beyond what the anchor alone achieves.

Effect of Reset Positional Encoding. We also ablate the Reset Positional Encoding by replacing it with standard absolute positional encoding. As shown in Figure 2, this leads to a substantial performance drop (23.8 vs. 33.5 BLEU at $t = 10$), indicating the necessity for the model to correctly identify token positions within each stream.

5.4 Encoder Choice and Anchor Quality

We further investigate whether the zero-shot transfer capability depends on the specific choice of multilingual encoder by comparing LaBSE against mBERT (Devlin et al., 2019) and XLM-R (Conneau et al., 2020).

Figure 3 demonstrates that the effectiveness of the semantic anchor relies on the specific proper-

ties of the multilingual encoder. When integrating LaBSE, mBERT, and XLM-R into the same PrismDiff architecture, all three encoders yield comparable performance on supervised En-X tasks. However, their zero-shot Es-X frontiers diverge drastically. While LaBSE maintains robust zero-shot performance of near 28 BLEU, both XLM-R and mBERT collapse to significantly lower performance levels (near 13 and 10 BLEU, respectively).

Importantly, we do not claim LaBSE is universally the best encoder choice across all decoding architectures. Experiments on AR models reveal that mBERT can achieve higher supervised quality (~ 39 BLEU) under traditional AR decoding, despite still failing catastrophically in Es-X zero-shot transfer (~ 8.5 BLEU). This suggests that the optimal encoder choice is dependent on both the decoding mechanism and the specific task. For PrismDiff, specifically, LaBSE is more effective because its language-agnostic sentence-level representations perfectly meet the need of a shared anchor to guide the iterative diffusion process.

5.5 Deployment Flexibility and Scaling

Our experiments further validate PrismDiff’s architectural flexibility for real-world deployment. First, the framework is inherently invariant to the generation order of target languages. For example, generating En- $\{\text{De}, \text{Fr}\}$ versus En- $\{\text{Fr}, \text{De}\}$ yields identical performance. Second, dynamically omitting target streams at inference time impacts translation quality by less than 1%, proving that the system allows for flexible configuration without re-training. A core motivation of our framework is efficient one-to-many generation. We analyze the latency and memory scaling behavior as the number of target languages (K) increases (see A.2). Traditional AR systems suffer from strict linear latency scaling, as generating K languages sequentially multiplies the temporal bottleneck. In contrast, although the canvas expands with more target streams, PrismDiff exhibits a sublinear latency growth thanks to the parallel diffusion refinement.

6 Conclusion

In this work, we explore the design space of parallel multilingual machine translation. To illustrate this, we propose PrismDiff, a discrete diffusion framework for parallel one-to-many translation. By guiding the generative process with a language-agnostic semantic anchor, PrismDiff en-

ables parallel one-to-many translation with high semantic consistency, controllable quality-latency trade-offs, and robust source-side zero-shot transfer. These results suggest that parallel multi-stream diffusion with a semantic anchor is a viable and flexible approach that offers a different quality-latency trade-off compared to AR systems. In supervised settings, PrismDiff with accelerated sampling achieves comparable translation quality to strong AR baselines while delivering a $2\times$ speedup. More notably, PrismDiff establishes a stronger zero-shot frontier: the semantic anchor design enables the framework to maintain approximately 75% of its supervised performance on unseen source languages, outperforming its AR counterparts in zero-shot BLEU.

Looking forward, our framework holds the potential to generalize to other one-to-many or many-to-many generation tasks beyond translation. Exploring more robust anchor alignment strategies could further improve performance.

Limitations

Despite its advantages in latency and flexibility, PrismDiff has several limitations.

First, as a diffusion-based NAR model, it does not uniformly outperform strong AR baselines on supervised translation quality. Its advantage lies in a controllable frontier: fewer refinement steps reduce latency, while additional steps enhance quality. Diffusion decoding also remains slower than single-step NAR models, necessitating further acceleration strategies.

Second, our current approach relies on a pretrained language-agnostic sentence encoder (LaBSE) to extract the semantic anchor. Although this anchor enables strong zero-shot transfer in PrismDiff, encoder ablations show that not every multilingual encoder provides a sufficiently aligned semantic space. A poorly represented language, domain, or a truly distant low-resource language in the pretrained encoder may greatly degrade transfer performance. Moreover, the best source representation can be decoder-dependent; different multilingual encoders may benefit different decoder architectures. Thus, our conclusion about LaBSE should be interpreted within PrismDiff rather than as a universal claim for all translation decoders.

Third, for experiment simplicity, we employ several vanilla implementation strategies of

MDMs, such as simple noise schedules and fixed language block lengths. Although our settings ($l_{\text{multi30k}} = 40$, $l_{\text{europarl}} = 80$) can cover most examples, there still exist longer sentences that will overflow, which considerably contributes to the poor performance on Europarl. Better performance may be achieved if these strategies and hyperparameters are more carefully tuned.

While exclusive attention already produces strong results through joint parameter optimization, exploring lightweight cross-stream interaction mechanisms, such as gated cross-attention between target streams, could be a promising direction but may also introduce more computation and inter-stream interference.

Additionally, our efficiency evaluation focuses on a standard single-GPU, single-instance, batch-size=1 latency, which closely reflects serving settings with extremely high latency requirements where synchronous batch is impractical. Optimizing batch throughput remains a direction to be explored in the future.

References

- Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. 2021. Structured denoising diffusion models in discrete state-spaces. *Advances in neural information processing systems*, 34:17981–17993.
- Ion Madrazo Azpiazu and Maria Soledad Pera. 2020. A framework for hierarchical multilingual machine translation. *arXiv preprint arXiv:2005.05507*.
- Grigory Bartosh, Teodora Pandeva, Sushrut Karmalkar, and Javier Zazo. 2026. Forward-learned discrete diffusion: Learning how to noise to denoise faster. *arXiv preprint arXiv:2605.18204*.
- Tiwei Bie, Maosong Cao, Kun Chen, Lun Du, Mingliang Gong, Zhuochen Gong, Yanmei Gu, Jiaqi Hu, Zenan Huang, Zhenzhong Lan, and 1 others. 2025. Llada2. 0: Scaling up diffusion language models to 100b. *arXiv preprint arXiv:2512.15745*.
- Linyao Chen, Aosong Feng, Boming Yang, and Zihui Li. 2023. Xdlm: Cross-lingual diffusion language model for machine translation. *arXiv preprint arXiv:2307.13560*.
- Xinhua Chen, Sitao Huang, Cong Guo, Chiyue Wei, Yintao He, Jianyi Zhang, Hai Li, Yiran Chen, and 1 others. 2025. Dpad: Efficient diffusion language models with suffix dropout. *arXiv preprint arXiv:2508.14148*.
- Zixiang Chen, Huizhuo Yuan, Yongqian Li, Yiwen Kou, Junkai Zhang, and Quanquan Gu. 2024. Fast

- sampling via discrete non-markov diffusion models with predetermined transition time. *Advances in Neural Information Processing Systems*, 37:106870–106905.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 8440–8451.
- Marta R Costa-Jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, and 1 others. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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)*, Minneapolis, Minnesota. Association for Computational Linguistics.
- Desmond Elliott, Stella Frank, Khalil Simaan, and Lucia Specia. 2016. Multi30k: Multilingual english-german image descriptions. In *Proceedings of the 5th Workshop on Vision and Language*, pages 70–74.
- Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, and 1 others. 2021. Beyond english-centric multilingual machine translation. *Journal of Machine Learning Research*, 22(107):1–48.
- Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic bert sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891.
- Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. 2019. 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)*, pages 6112–6121.
- Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and Lingpeng Kong. 2023. Diffuseq: Sequence to sequence text generation with diffusion models. In *International Conference on Learning Representations (ICLR 2023)(01/05/2023-05/05/2023, Kigali, Rwanda)*.
- Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376.
- J Gu, J Bradbury, C Xiong, VOK Li, and R Socher. 2018. Non-autoregressive neural machine translation. In *International Conference on Learning Representations (ICLR)*.
- Yiwen Guan and Jacob Whitehill. 2025. Transformer-encoder trees for efficient multilingual machine translation and speech translation. *arXiv preprint arXiv:2509.17930*.
- Junliang Guo, Xu Tan, Di He, Tao Qin, Linli Xu, and Tie-Yan Liu. 2019. Non-autoregressive neural machine translation with enhanced decoder input. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3723–3730.
- Zhengfu He, Tianxiang Sun, Qiong Tang, Kuanning Wang, Xuan-Jing Huang, and Xipeng Qiu. 2023. Diffusionbert: Improving generative masked language models with diffusion models. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 4521–4534.
- Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851.
- Ross Irwin, Alessandro Tibo, Jon Paul Janet, and Simon Olsson. 2025. Semlaflow—efficient 3d molecular generation with latent attention and equivariant flow matching. In *International Conference on Artificial Intelligence and Statistics*, pages 3772–3780. PMLR.
- Melvin Johnson, Mike Schuster, Quoc Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, and 1 others. 2017. Googles multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.
- Jungo Kasai, Nikolaos Pappas, Hao Peng, James Cross, and Noah A Smith. 2020. Deep encoder, shallow decoder: Reevaluating non-autoregressive machine translation. *arXiv preprint arXiv:2006.10369*.
- Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *The Tenth Machine Translation Summit Proceedings of Conference*, pages 79–86. International Association for Machine Translation.
- Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. 2022. Diffusion-lm improves controllable text generation. *Advances in neural information processing systems*, 35:4328–4343.

Zhiyuan Liu, Yicun Yang, Yaojie Zhang, Junjie Chen, Chang Zou, Qingyuan Wei, Shaobo Wang, and Linfeng Zhang. 2025. d1lm-cache: Accelerating diffusion large language models with adaptive caching. *arXiv preprint arXiv:2506.06295*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, JUN ZHOU, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. 2025. Large language diffusion models. In *ICLR 2025 Workshop on Deep Generative Model in Machine Learning: Theory, Principle and Efficacy*.

Matt Post. 2018. A call for clarity in reporting bleu scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, page 186. Association for Computational Linguistics.

Lihua Qian, Hao Zhou, Yu Bao, Mingxuan Wang, Lin Qiu, Weinan Zhang, Yong Yu, and Lei Li. 2021. Glancing transformer for non-autoregressive neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1993–2003.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. Comet: A neural framework for mt evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Chengyue Wu, Hao Zhang, Shuchen Xue, Zhijian Liu, Shizhe Diao, Ligeng Zhu, Ping Luo, Song Han, and Enze Xie. 2025. Fast-d1lm: Training-free acceleration of diffusion llm by enabling kv cache and parallel decoding. *arXiv preprint arXiv:2505.22618*.

Yisheng Xiao, Lijun Wu, Junliang Guo, Juntao Li, Min Zhang, Tao Qin, and Tie-yan Liu. 2023. A survey on non-autoregressive generation for neural machine translation and beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(10):11407–11427.

Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. 2025. Dream 7b: Diffusion large language models. *arXiv preprint arXiv:2508.15487*.

Chia-Kai Yeh, Chih-Chun Chen, Ching-Hsien Hsu, and Jen-Tzung Chien. 2024. Cross-modality diffusion modeling and sampling for speech recognition. In *INTERSPEECH*.

A Appendix

A.1 Logarithm-Uniform Masking Schedule

Let L denote the maximum sequence length. The masking ratio m is bounded within $[1/L, 1]$, where $1/L$ corresponds to the single-token mask limit. At each refinement step, tokens are remasked based on predicted confidence scores: the model retains the top $(1 - m) \times L$ tokens with highest predicted probability and remasks the remainder, where m is determined by the schedule. We define a log-uniform distribution over m by sampling:

$$\log m \sim \mathcal{U}\left(\log \frac{1}{L}, 0\right)$$

This formulation assigns equal probability mass to equal intervals on the logarithmic scale. To adapt this continuous prior for fixed-step training, we discretize the schedule over the total training steps N . In practice, the log-uniform schedule is implemented deterministically by mapping training steps to masking ratios. For example, if using $L = 40$ with $K = 4$ discrete levels, the corresponding time step sequence t_i is given by:

$$t_i = \exp\left(\log\left(\frac{1}{L}\right) \times i\right) \times N$$

, which are:

$$t_0 = N, t_1 = 0.4N, t_2 = 0.16N, t_3 = 0.063N$$

This ensures that the training process effectively covers the full range of masking ratios with the desired log-uniform distribution. We found this masking schedule consistently outperforms other simple strategies in our task.

A.2 Scaling Analysis: Latency and Memory

A detailed latency–language scaling analysis is provided in Figure 4, comparing decoder-only AR (8-layer and 4-layer) against PrismDiff ($T = 100$, $t = 50$, and $t = 25$). PrismDiff exhibits sublinear latency scaling with respect to the number of target streams K . With exclusive attention, each refinement step requires one forward pass over the full canvas of size L , where each stream attends only to itself and the anchor with cost $O(L^2)$. The per-step cost is therefore $O(K \cdot L^2)$, and the total cost over t diffusion refinement steps is $O(t \cdot K \cdot L^2)$. Generating K languages in a single forward pass therefore is $K \times$ the cost of a single-language diffusion pass, but $K \times$ cheaper than running K separate passes.

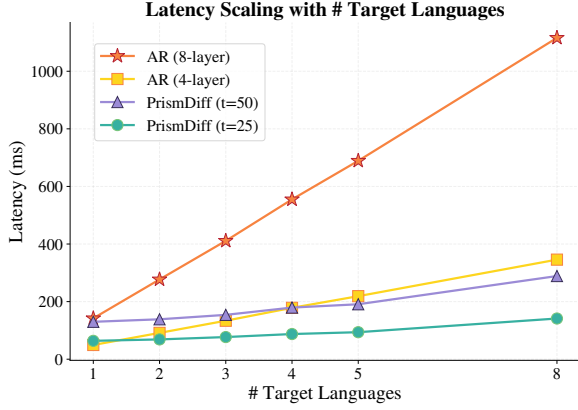


Figure 4: **Latency–language scaling graph.** The PrismDiff models are trained with $T = 100$, and accelerated sampling is performed during the inference stage using $t = 50$ or $t = 25$.

In contrast, AR systems decoding for K languages requires K sequential passes, each with $O(L^2)$ attention cost over L autoregressive steps, resulting in $O(K \cdot L^3)$ in total. PrismDiff’s total cost of $O(t \cdot K \cdot L^2)$ is therefore favorable when $t \ll L$. We note that AR systems could reduce wall-clock latency for K targets by running K model replicas simultaneously. However, this would multiply the memory consumption by K whereas PrismDiff generates all K targets with a single model instance.

Block length	Number of Target Languages (K)					
	1	2	3	4	5	8
$l = 40$	2.66	2.70	2.73	2.77	2.81	2.91
$l = 100$	2.72	2.81	2.90	2.99	3.08	3.35
$l = 300$	2.90	3.17	3.44	3.71	3.98	4.80

Table 3: **Peak GPU memory footprint** (in GB) of PrismDiff during parallel multi-stream inference. The memory scales gracefully even under extreme stress tests (e.g., $K = 8$ targets with $l = 300$ tokens).

Table 3 reports peak GPU memory footprint (GB) of PrismDiff under different block lengths (l) and number of target streams (K). With $l = 40$ block length, memory increases by less than 10% as K scales from 1 to 8 (2.66 GB to 2.91 GB), confirming that the canvas-based multi-stream design does not incur significant memory overhead in practice. The memory growth becomes more pronounced at larger block lengths (e.g., $l = 300$), which is expected as the total canvas size scales as $L_{\text{src}} + K \times L_{\text{tgt}}$. The memory scales gracefully even under extreme stress tests (e.g., $K = 8$

with $L = 300$), confirming that while the hyper-sequence design inherently costs more memory for long texts, it easily fits in the capacity of standard GPUs without requiring complex multi-GPU model parallelism. Note that AR memory does not scale with K since each language is decoded sequentially; for reference, an 8-layer decoder-only AR model on Multi30K (max length < 40) has a peak memory of approximately 2.69 GB, comparable to PrismDiff at $K = 1$ (2.66 GB). The key difference lies not in memory but in latency: the cost of AR models is reflected in a linear latency growth, as shown in Figure 4.

A.3 Full Results

Full experimental results on Multi30K are reported in Tables 5 – 8. The results include comparing different implementation setups and architectures (Table 5), using different sampling budgets (Table 6), decoder-only AR baselines with various numbers of decoder layers (Table 7), and applying different multilingual encoders for extracting source-side representation (anchor) as the input of PrismDiff (Table 8).

A.4 The Role of Cross-Lingual Alignment in Attention

The performance gap of Diff-Indep using full attention versus using exclusive attention suggests that full cross-stream attention plays different roles depending on the conditioning signal. When conditioned directly on source tokens (Diff-SrcTokens), the model lacks a centralized, language-agnostic representation. Without a semantic anchor, full attention can partially compensate for weak source conditioning: each target stream essentially treats the noisy states of other streams as auxiliary context, forming a pseudo multi-source prediction problem to extract semantic clues. However, when PrismDiff is conditioned on a cross-lingually aligned semantic anchor, direct target-stream attention can introduce surface-form interference across languages. Therefore, exclusive stream attention encourages each language to perform its own denoising process, while semantic coordination is pre-solved by the shared anchor.

A.5 Inference Examples

One example of the accelerated inference process of PrismDiff trained on Multi30K with a total diffusion steps of $T = 100$ is shown in Table 9, and

Table 10 contains another example trained on Europarl with $T = 40$. All target languages are refined simultaneously and generated together at each step. The canvas size and the refinement steps of each target language is truncated in the examples for better readability. A larger step t means an earlier stage in the inference time, so more mask tokens ([M]) will be presented in the example. The translation displayed in the last step ($t = 0$) is CTC-collapsed $t = 0$ prediction, which removes repetitive tokens (only if there exist).

A.6 Zero-Shot Transfer Examples

We present representative examples illustrating typical failure modes of AR systems under zero-shot transfer in Table 11. We provide a qualitative analysis using zero-shot Spanish-to-French (Es $\rightarrow$ Fr) translation examples from Multi30K. The original task on which the model is trained on is En{De, Fr}. In Example 1, the AR model suffers from error cascading and hallucination. In Example 2 and Example 3, the AR baselines exhibits structural collapse, generating non-existing words.

A.7 Implementation of TET

Our implemented TET model on Europarl contains 4 leaf nodes (corresponding to the target languages), each has a depth of 6. We list the detailed paths of all languages in TET in Table 4. In all, the TET model contains 16 Transformer encoder layers, resulting in a total parameter count of 556M.

TET’s extremely low latency is attributed to its single-pass CTC decoding. Unlike sequential autoregressive decoding, or iterative diffusion refinements, TET generates all target languages in a single forward pass. The higher parameter count does not proportionally increase latency due to the parallel tree structure.

Language	Path
Dutch	–Germanic–Western–Dutch ₁ –Dutch ₂
Danish	–Germanic–Northern–Danish ₁ –Danish ₂
French	–Romance–Western–French ₁ –French ₂
Romanian	–Romance–Eastern–Romanian ₁ –Romanian ₂

Table 4: Paths of all languages in TET on Europarl. All paths begin with "*common*₁–*common*₂– $\dots$ ", which is omitted in this table for simplicity.

dataset is publicly available as open data and widely permitted for research purposes. We utilize the pre-trained LaBSE (Language-Agnostic BERT Sentence Embeddings) and mBERT (multilingual BERT for model design and evaluation, both of which are officially released by Google and distributed under the Apache License 2.0. We also employ XLM-R (XLM-RoBERTa), developed by Meta, which is released under the MIT License. We also use the NLLB (No Language Left Behind, e.g., NLLB-200) model, which is distributed by Meta under the CC-BY-NC 4.0, strictly limiting its use to non-commercial research purposes. We confirm that our use of the datasets is consistent with their intended use for machine translation research. Similarly, all models are employed for feature extraction and data synthesis, respectively, aligning with their original research purposes.

A.8 Licenses and Usage of Artifacts

The Multi30K dataset is publicly available and licensed under the CC BY-SA 4.0. Europarl

Src.	Model	Architecture	Attn.	T	BLEU		COMET		Avg.		Latency (ms)
					De	Fr	De	Fr	BLEU	COMET	
En	Diff-Indep			10	28.47	39.51	0.62	0.73	33.99	0.68	26.9
				100	31.07	40.74	0.66	0.76	35.91	0.71	254.5
	Diff-SrcTokens	- CTC		full 10	30.35	37.32	0.61	0.70	33.84	0.66	25.6
				full 10	29.05	36.31	0.58	0.67	32.68	0.63	24.5
				full 100	32.12	40.76	0.65	0.74	36.44	0.70	234.4
				excl. 100	30.95	38.64	0.65	0.74	34.80	0.695	247.4
	AR	enc-dec			29.90	35.83	0.67	0.73	32.87	0.70	88.9
					20.85	22.65	0.61	0.66	21.75	0.64	90.1
					32.82	41.67	0.68	0.76	37.25	0.72	148.3
	PrismDiff			full 10	27.91	37.40	0.61	0.71	32.66	0.66	27.8
				full 10	27.16	36.62	0.58	0.68	31.89	0.63	32.5
				excl. 10	28.48	38.47	0.62	0.72	33.48	0.67	26.5
		- CTC		excl. 10	27.29	37.09	0.59	0.70	32.19	0.65	26.7
				excl. 10	19.56	28.04	0.53	0.62	23.80	0.58	26.2
		Abs. PE		full 100	31.92	40.24	0.67	0.75	36.08	0.71	259.0
				excl. 100	33.16	41.10	0.68	0.76	37.13	0.72	278.3
Es	Diff-SrcTokens			full 10	0.28	0.16	0.28	0.29	0.22	0.29	21.9
				full 100	0.14	0.31	0.29	0.30	0.23	0.30	234.0
				excl. 100	0.30	0.31	0.26	0.32	0.31	0.29	238.6
	AR	dec-only + Z			20.48	29.95	0.60	0.71	25.22	0.66	154.5
	PrismDiff			full 10	19.43	30.43	0.56	0.68	24.93	0.62	29.5
				excl. 10	20.33	31.51	0.57	0.69	25.92	0.63	27.2
		Abs. PE		excl. 10	14.75	21.53	0.50	0.59	18.14	0.55	25.8
				full 100	22.37	33.13	0.62	0.73	27.75	0.67	25.6
				excl. 100	23.25	33.17	0.62	0.73	28.21	0.68	284.0

Table 5: Results on Multi30K. In architecture ablations, “enc-dec” is encoder-decoder and “dec-only” denotes decoder-only; “+Z” means using semantic anchor Z as input, otherwise the AR model directly uses source tokens as input; “-CTC” means without CTC decoding; “Abs. PE” is using absolute positional encoding instead of reset positional encoding. “Attn.” represents the attention mode used by PrismDiff, in which “full” means using full attention across target language, and “excl.” means using exclusive attention for each target language. T is the total diffusion steps the model is trained on, and all diffusion systems report the results using T full sampling steps with uniform noise scheduler. “Src.” lists the source language input to the model, in which Spanish (Es) is a zero-shot source language that the model is never trained on.

Src.	Sampl. Steps (t)	BLEU		COMET		Avg.		Latency (ms)
		De	Fr	De	Fr	BLEU	COMET	
En	100	33.16	41.10	0.68	0.76	37.13	0.72	278.3
	50	32.90	41.22	0.68	0.76	37.06	0.72	147.2
	30	33.20	40.98	0.67	0.76	37.09	0.72	83.2
	25	32.99	40.95	0.67	0.76	36.97	0.72	68.7
	15	32.30	40.01	0.67	0.75	36.16	0.71	40.9
Es	100	23.25	33.17	0.62	0.73	28.21	0.68	284.0
	50	23.44	33.01	0.62	0.73	28.23	0.67	141.7
	30	23.69	32.91	0.62	0.72	28.30	0.67	83.1
	25	23.43	33.01	0.62	0.72	28.22	0.67	69.3
	15	22.91	32.34	0.61	0.71	27.63	0.66	40.7

Table 6: Sampling steps vs. performance on Multi30K. The PrismDiff model is trained with a total diffusion steps of $T = 100$, and is evaluated under different accelerated sampling steps (t) using logarithm-uniform noise scheduler.

<i>Src.</i>	Num. Layers	BLEU		COMET		Avg.		Latency (ms)
		De	Fr	De	Fr	BLEU	COMET	
<i>En</i>	8	32.82	41.67	0.68	0.76	37.25	0.720	148.3
	6	32.36	40.45	0.70	0.77	36.41	0.735	120.9
	4	31.97	40.08	0.68	0.76	36.03	0.720	92.5
	3	30.79	40.16	0.67	0.75	35.48	0.710	76.0
	2	30.96	38.77	0.66	0.75	34.87	0.705	58.6
<i>Es</i>	8	20.48	29.95	0.60	0.71	25.22	0.655	154.5
	6	20.98	30.32	0.63	0.73	25.65	0.678	121.4
	4	21.67	30.39	0.62	0.72	26.03	0.670	92.5
	3	19.73	29.35	0.60	0.71	24.54	0.655	77.6
	2	19.30	28.82	0.59	0.70	24.06	0.645	61.0

Table 7: Number of AR decoder layers vs. performance on Multi30K.

<i>Src.</i>	Encoder	Sampl. Steps (<i>t</i>)	BLEU		COMET		Avg.		Latency (ms)
			De	Fr	De	Fr	BLEU	COMET	
<i>En</i>	LaBSE	100	33.16	41.10	0.68	0.76	37.13	0.72	376.8
		30	33.20	40.98	0.67	0.76	37.09	0.72	83.2
		25	32.99	40.95	0.67	0.76	36.97	0.72	68.7
		15	32.30	40.01	0.67	0.75	36.16	0.71	40.9
	mBERT	100	33.44	40.55	0.67	0.75	37.00	0.71	281.2
		30	33.00	39.96	0.67	0.75	36.48	0.71	82.9
		25	32.64	40.04	0.67	0.75	36.34	0.71	69.4
		15	32.34	39.11	0.66	0.74	35.73	0.70	40.1
	XLM-R	100	33.45	40.20	0.68	0.76	36.83	0.72	286.0
		30	33.43	40.09	0.67	0.76	36.76	0.72	86.8
		25	33.52	39.84	0.67	0.76	36.68	0.72	72.8
		15	31.62	39.00	0.66	0.75	35.31	0.71	44.6
<i>Es</i>	LaBSE	100	23.25	33.17	0.62	0.73	28.21	0.68	284.0
		30	23.69	32.91	0.62	0.72	28.30	0.67	83.1
		25	23.43	33.01	0.62	0.72	28.22	0.67	69.3
		15	22.91	32.34	0.61	0.71	27.63	0.66	40.7
	mBERT	100	10.10	16.06	0.47	0.54	13.08	0.51	283.5
		30	9.79	15.93	0.47	0.53	12.86	0.50	82.8
		25	10.13	15.87	0.47	0.53	13.00	0.50	69.2
		15	9.63	15.00	0.46	0.52	12.32	0.49	40.0
	XLM-R	100	13.68	18.18	0.51	0.59	15.93	0.55	289.5
		30	13.25	17.67	0.51	0.59	15.46	0.55	87.1
		25	13.04	17.87	0.51	0.59	15.46	0.55	71.7
		15	12.74	17.46	0.51	0.58	15.10	0.55	44.2

Table 8: Performance comparison of using different multilingual encoders to extract the source-side representation (anchor) for PrismDiff on Multi30K. All PrismDiff models are trained with total diffusion steps of $T = 100$.

Step	German	French
t=54.1	Ein [M] [M] [M] [M] [M] [M] [M] [M] [M] [M] [M]	Un homme [M] [M] [M] [M] [M] [M] [M] [M] [M] [M] [M]
t=18.4	Ein Mann schläft in einem grünen [M] auf [M] [M]fa [M]	Un homme dormant dans une [M] [M] un can [M]é [M] [M]
t=5.4	Ein Mann schläft in einem grünen [M] auf einem Sofa.	Un homme dormant dans une pièce sur un canapé [M].
t=0	Ein Mann schläft in einem grünen Zimmer auf einem Sofa.	Un homme dormant dans une pièce sur un canapé vert.
Groundtruth	Ein Mann schläft in einem grünen Raum auf einer Couch.	Un homme dormant dans une chambre verte sur un canapé.

Table 9: An example of accelerated (with 25 refinement steps) translation process of PrismDiff ($T = 100$) trained on Multi30K. The canvas size and the number of the steps are reduced in this example for better readability. The “t=0” is the result after CTC collapse. The source English sentence is: “A man sleeping in a green room on a couch.”

Step	French	Romanian	Danish
t=30	Le Parlement européen [M] [M] [M] [M] [M] [M] [M] [M] [M] [M] [M] [M] [M] [M]	Parlamentul European a votat [M] [M] [M] [M] [M] [M] [M] [M] [M] [M] [M] [M] [M] [M]	[M] [M] [M] [M] [M] [M] [M] [M] [M] [M] [M] [M] [M] [M] [M] [M] [M] [M] [M] [M]
t=20	Le Parlement européen a voté [M] [M] résolution [M] [M] [M] [M] [M] [M] [M] [M]	Parlamentul European a votat o rezolu [M] [M] [M] [M] [M] [M] [M] [M] [M] [M] [M] [M]	Parlamentet har stemt for [M] beslutning om [M] [M] [M] [M] [M] [M] [M] [M] [M] [M] [M]
t=10	Le Parlement européen a voté pour [M] résolution [M] l’avenir du Fonds social [M] [M]	Parlamentul European a votat [M] rezolu [M] referitoare la viitorul Fondului [M] [M] [M]	Parlamentet har stemt for en beslutning om Den Europæiske Social [M] [M] [M] [M] [M]
t=0	Le Parlement européen a voté pour une résolution sur l’avenir du Fonds social européen.	Parlamentul European a votat o rezoluie referitoare la viitorul Fondului social european.	Parlamentet har stemt for en beslutning om Den Europæiske Socialfonds fremtid.
Groundtruth	Le Parlement européen a voté une résolution sur l’avenir du Fonds social européen.	Parlamentul European a votat pentru o rezoluie referitoare la viitorul Fondului social european.	Parlamentet har stemt for beslutningen om Den Europæiske Socialfonds fremtid.

Table 10: An example of translation process of PrismDiff ($T = 40$) trained on Europarl. Here, only 3 target languages are displayed for simplicity. The canvas size and the number of the steps are reduced in this example for better readability. The “t=0” is the result after CTC collapse. The source English sentence is: “The European Parliament has voted for a resolution on the future of the European Social Fund.”

Example 1	
Source (En)	<i>A child is splashing in the water</i>
Source (Es)	<i>Un niño está salpicando en el agua</i>
Groundtruth (Fr)	"Un enfant se plonge dans l'eau"
AR	<i>Un enfant est en train de donner de l'eau en train de shooter dans l'eau</i>
PrismDiff	<i>Un enfant fait des éclaboussures dans l'eau</i>
Example 2	
Source (En)	<i>This lady has heard a funny joke and laughing</i>
Source (Es)	<i>Esta señora ha oído un chiste divertido y riéndose</i>
Groundtruth (Fr)	<i>Cette dame a entendu une blague drôle et ri</i>
AR	<i>Cette femme a une ribe riée et rit riant</i>
PrismDiff	<i>Cette femme écoute une drôle de rire et rire</i>
Example 3	
Source (En)	<i>A man holding up another with his back</i>
Source (Es)	<i>Un hombre sosteniendo a otro con la espalda</i>
Groundtruth (Fr)	<i>Un homme qui tient un autre avec le dos</i>
AR	<i>Un homme tient un autre ab qui avec l'arrière</i>
PrismDiff	<i>Un homme se tenant un autre avec le dos</i>

Table 11: Zero-shot examples on Multi30K. The input source is Spanish (Es), and target languages are German (De) and French (Fr). We only show French sentences for illustration. The English source sentences in this table are only for reference (not used as input).