Benchmarking Diffusion Models for Machine Translation

Yunus Demirag, Danni Liu, Jan Niehues
Karlsruhe Institute of Technology, Germany
yunus.demirag@student.kit.edu, {danni.liu, jan.niehues}@kit.edu

Abstract

Diffusion models have recently shown great potential on many generative tasks. In this work, we explore diffusion models for machine translation (MT). We adapt two prominent diffusion-based text generation models, Diffusion-LM and DiffuSeq, to perform machine translation. As the diffusion models generate non-autoregressively (NAR), we draw parallels to NAR machine translation models. With a comparison to conventional Transformer-based translation models, as well as to the Levenshtein Transformer, an established NAR MT model, we show that the multimodality problem that limits NAR machine translation performance is also a challenge to diffusion models. We demonstrate that knowledge distillation from an autoregressive model improves the performance of diffusion-based MT. A thorough analysis on the translation quality of inputs of different lengths shows that the diffusion models struggle more on long-range dependencies than other models.

1 Introduction

Diffusion models have shown promising results in a wide range of generative tasks, such as image generation (Ho et al., 2020; Nichol et al., 2022), text-to-speech synthesis (Jeong et al., 2021), and robotic control (Chi et al., 2023), but their application to natural language processing (NLP) is still a less explored direction. The last two years have seen various approaches to this (Zou et al., 2023), including discrete (token level) diffusions (Reid et al., 2022) and continuous (embedded) diffusions. Continuous diffusion models typically generate whole sequences in an iterative and non-autoregressive (NAR) manner, and have shown strong results for controllable generative modelling (Li et al., 2022; Chen et al., 2023). They have also been applied to sequence-to-sequence tasks such as open-domain dialog and question generation (Gong et al., 2023; Yuan et al., 2022). In this work, we focus on machine translation (MT), another sequence-to-sequence task that requires fluent outputs over a vocabulary different from the input and the preservation of semantic meanings of the input sequences. Despite potential speed advantages[1], NAR translation models tend to lag behind their AR counterparts in translation quality[2] (Libovický and Helcl, 2018; Gu et al., 2019; Gu and Kong, 2021; Kasai et al., 2021) as a result of the *conditional independence assumption*, where output tokens are generated independent of each other. This prompts us to compare diffusion-based MT models to conventional NAR MT models. We explore how techniques commonly applied to NAR MT models could benefit diffusion-based models. Specifically, we seek to answer the following questions: **1)** How can we adapt existing diffusion-based text generation models to machine translation? **2)** How do these diffusion-based MT models compare to standard AR and NAR machine translation models? **3)** What are reasons for the performance gap and how can we bridge the gap?

2 Background and Related Work

2.1 Diffusion Models

While there exist many other families of diffusion models[3], we limit our discussion on the denoising diffusion probabilistic model (DDPM) (Ho et al., 2020), which can be viewed as a variational diffusion model (Sohl-Dickstein et al., 2015; Ho et al., 2020; Kingma et al., 2021). In general terms, a diffusion model is a type of generative model that learns to model the probability distribution of given datasets. Its essential components are: 1) the *forward process* in which noise is iteratively added

[1] which has been called into question under realistic conditions (Helcl et al., 2022)

[2] Some recent exceptions include Qin et al. (2022) based on hybrid NAR and AR generation.

[3] We refer interested readers to Luo (2022) for a more general coverage.

Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop, pages 313–324

to the data, i.e., the data is diffused for a given number of time steps; 2) a predefined *noise schedule* which determines the amount noise added at every time step; 3) the parametric *backward process* that is optimized to match the time-reverse forward process, thereby recreating the data sample. Specifically, the stochastic model consists of $T+1$ random variables with T indicating the number of time steps. These random variables include the observation variable $\mathbf{X}_0$ and T latent variables $\mathbf{X}_1, \ldots, \mathbf{X}_T \in \mathbb{R}^d$. Between them, we assume a conditional probability distribution with some regularity constraints[4], commonly a normal distribution with the mean and variance being dependent on the previous state. The process is illustrated with the light gray nodes in Figure 1.

Forward Process In the forward process, the observed information is diffused by the conditional probability adding a small amount of noise in each step according to the *noise schedule* $(\alpha_i)_{i=1}^T$ where α_i defines the noise applied in the ith time step[5]. Therefore, the forward process is a time-discrete stochastic process, which can be described by $q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t|\sqrt{\alpha_i}\mathbf{x}_{t-1}, (1-\alpha_i)I_d)$ for $t = 2, \ldots, T$ (Li et al., 2022; Luo, 2022; Ho et al., 2020). Utilizing the formula for conditional multivariate normal distributions, we can derive

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)=N\Big(\mathbf{x}_{t-1}|\mu(\mathbf{x}_t,\mathbf{x}_0),\tfrac{(1-\bar{\alpha}_{t-1})(1-\alpha_t)}{1-\bar{\alpha}_t}I_d\Big). \tag{1a}$$

where

$$\mu(\mathbf{x}_t,\mathbf{x}_0):=\tfrac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\mathbf{x}_t+\sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\mathbf{x}_0}{1-\bar{\alpha}_t}. \tag{1b}$$

Backward Process The Markovian backward process is defined as $p(\mathbf{x}_{t-1}|\mathbf{x}_t) = q(\mathbf{x}_{t-1}|\mathbf{x}_t, \hat{\mathbf{x}_0}(\mathbf{x}_t))$ using a neural network $\hat{\mathbf{x}_0}(\mathbf{x}_t)$ to estimate the initial data $\mathbf{x}_0$ in every step (Li et al., 2022). Sampling from the model corresponds to first sampling $\mathbf{X}_T \sim \mathcal{N}(0, 1)$ and then sampling a backward trajectory in an iterative manner. An example trajectory is illustrated in Appendix A. Accordingly the Evidence Lower Bound (ELBO) of $\log p_\theta(\mathbf{x}_0)$ for training data $\mathbf{x}_0$ is used as a loss function (Luo, 2022).

2.2 Diffusion Models for Language Modeling

Language modeling is the task of assigning probabilities to sequences of words $y_{1,\ldots,n}$ and is a central task in NLP. Here we describe two prominent approaches of using diffusion models for language modeling: Diffusion-LM (Li et al., 2022) and DiffuSeq (Gong et al., 2023). Although similar in principle, DiffuSeq uses *classifier-free guidance* (Ho and Salimans, 2022) to model a conditional diffusion process for sequence-to-sequence tasks.

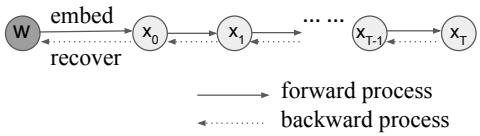

Figure 1: An illustration of the forward and backward diffusion processes for text generation.

2.2.1 Diffusion-LM

The model underlying Diffusion-LM (Li et al., 2022) is similar to the DDPM (Ho et al., 2020) proposed for image generation. A main difference is the extra requirement of handling text outputs, which are *discrete* in nature unlike images. This calls for two modifications illustrated on the left-hand-side of Figure 1: when *embedding* the text as training targets, and when *recovering* the discrete tokens from continuous diffusion states.

Embedding Function To embed discrete text tokens, a word embedding lookup table is used as in many other NLP models. This means the embedding function E_θ is simply a context-free token-wise embedding. It is only used to obtain the training targets during training and when filling in masked data. The embedding vectors are optimized end-to-end together with the backward process, as Li et al. (2022) found pretrained word embeddings degraded performance.

Recovery Function When generating the text outputs, i.e., mapping from continuous diffusion states to discrete tokens, the recovery function R_θ is a linear layer followed by a softmax activation, like the output layer in most NLP models. It can be viewed as a nearest neighbour lookup in the embedding space. Like the word embeddings, the weights of the recovery function are also trained jointly with the diffusion model. Moreover, as recovering from the diffusion states to single word embeddings (i.e., committing) is often difficult, Li et al. (2022) proposed a *clamping trick* to force the model to commit to certain word embeddings at intermediate diffusion steps. Specifically, this is achieved by mapping the predicted initial data

[4]Specifically a Markov kernel in the mathematical sense.

[5]For which $\bar{\alpha}_T \simeq 0$, where $\forall i \in \{1, \ldots, T\}$: $\bar{\alpha}_i := \prod_{t=1}^{i} \alpha_t$ and $\alpha_i \in (0, 1)$ needs to hold.

$\hat{\mathbf{x}}_0(\mathbf{x}_t)$ to the closest word embedding sequence at each time step.

Classifier Guidance Although Diffusion-LM can be used as a language model in general, its main focus is controllable text generation (Li et al., 2022), where the backward process is modified for the end result to satisfy one or multiple control targets, such as sentiment or syntactic structure. In most experiments[6] by Li et al. (2022), control is achieved by *classifier guidance*, i.e., training a classifier to model $\mathbf{P}(\cdot|\mathbf{X}_t)$ on the diffusion latent variables $\mathbf{X}_t$, and running gradient updates $\nabla_{\mathbf{X}_t} \log \mathbf{P}(\text{desired class}|\mathbf{X}_t)$ at each step during the backward process. The text generation process is thereby guided towards desired classes.

Infilling Procedure For the task of filling in missing data, e.g., sentence completion based on sounding sentences, Diffusion-LM uses the *infilling algorithm*. This approximates conditional distributions where the variable we want to condition on is already modelled by the diffusion model, and is comparable to the image inpainting capability (Lugmayr et al., 2022) of diffusion models for image generation. To achieve this, the conditioning information is kept fixed at its desired value throughout the backward process.

2.2.2 DiffuSeq

Difference to Diffusion-LM Unlike Diffusion-LM which focuses on controllable generation, DiffuSeq (Gong et al., 2023) focuses on sequence-to-sequence tasks, and the authors argue that classifier guidance is insufficient for this type of task, since the fine-grained input-output relation cannot be achieved by a finite number of classifiers. The authors therefore propose a *classifier-free* approach.

Classifier-Free Diffusion Bypassing classifier guidance, DiffuSeq (Gong et al., 2023) directly models the transformation between (source $\oplus$ random) and (source $\oplus$ target) where $\oplus$ indicates the concatenation operation. Specifically, DiffuSeq models the distribution of the target sequence conditioned by the source sequence. To achieve that, DiffuSeq used *conditional noising*, which only applies noise to the target sequence while leaving the source sequence fixed. This is done both in training and sampling/decoding. The sampling procedure is analogous to the infilling procedure of Diffusion-LM as a result.

[6]One exception out of their 6 setups is the infilling experiment, which does not need a classifier.

2.3 Non-Autoregressive Models and the Multimodality Problem

Non-autoregressive Transformer models (NAT) (Gu et al., 2018) are based on the conditional independence assumption, where the generation of tokens in the target sequence does not depend on each other. While allowing for a rapid decoding process, this introduces the *multimodality problem* (Gu et al., 2018) due to nondeterminism in the dataset. Nondeterminism in the dataset can be explained by the example of German sentences "*Danke schön*" and "*Vielen Dank*" both being possible translations of "Thank you", but a model following the conditional independence assumption cannot allow both variants (Gu et al., 2018). Diffusion models do not follow the conditional independence assumption, so it is unclear whether the nondeterminism in the dataset impacts model performance.

3 Adapting Diffusion Models to Machine Translation

Machine translation is an instance of the conditional language modeling problem. Specifically, it aims to automatically translate text from one *source* language to another *target* language, and may be described as modeling the distribution over the target space conditioned by a sequence from the source space. Currently, the primary model choice for machine translation is the encoder-decoder architecture, especially the Transformer (Vaswani et al., 2017), where an encoder module first encodes the source sequence, passing the encoding on to the decoder, which autoregressively generates an output sequence conditioned by the source encoding.

To this end, formally we describe the probability of a sequence y given the conditioning information x under the transformer model p_θ:

$$p_\theta(y;x) = \prod_{i=1}^{|y|} \underbrace{p_\theta(y_i|y_1,\ldots,y_{i-1};x)}_{\text{modelled explicitly}}. \quad (2)$$

Considering that the diffusion models described in §2.2 generate sequences en bloc, we constrain our problem to only consider pairs of sequences of a combined maximum length S. Accordingly, sequences are padded or truncated to the length S.

3.1 Diffusion-LM for Machine Translation

Reasons for a Classifier-Free Approach As introduced in §2.2.2, the source-target transforma-

tion required for machine translation is more complex than controllable generation guided by discrete classes. Specifically, it requires the model to safeguard against alterations in semantic meaning and demonstrate the ability to pay close attention to different words in the source sequence depending on the token in the target sequence. So for a classifier guidance approach, one could potentially train a Transformer model to back-translate from target to source, and use gradients from this model to guide the generation. However, as the generation output is highly dependent on the guiding model, it remains questionable whether this approach provides any benefits over an autoregressive Transformer model. This motivated us to approach diffusion-based machine translation by classifier-free guidance.

Approach We use a shared dictionary $V = V_s, V_t$, and seek to model the joint distribution $\mathbf{J} : \mathcal{P}(V^S) \rightarrow [0, 1]$ of pairs of source and target sequences by training Diffusion-LM on this task. Given a set of training source and target pairs $(s^{(1)}, t^{(1)}), \ldots, (s^{(n)}, t^{(n)})$, we use the concatenated source-target sequences, where $j^{(i)} = s^{(i)} \oplus (\hat{s}) \oplus t^{(i)}$ for $i \in [1, n]$. The source and target sequences are separated by a reserved separator token $\hat{s} \in V$. A Diffusion-LM model is then trained to maximize the likelihood of the training sequences $j^{(1)}, \ldots, j^{(n)}$. By using the infilling algorithm to approximate the conditional distribution of the target sequence given the source sequence, translation is then performed without relying on classifier guidance.

3.2 DiffuSeq for Machine Translation

As DiffuSeq is proposed for sequence-to-sequence tasks, we can directly apply it on machine translation. Like the Diffusion-LM-based model, the DiffuSeq-based models use shared vocabularies $V_s, V_t = V$. The sampling algorithm is the same as the infilling algorithm for Diffusion-LM (Gong et al., 2023).

3.3 Sequence-Level Knowledge Distillation

Motivated by theories and findings in the machine translation and linguistics literature, we proceed to improve diffusion-based translation models.

To tackle the multimodality problem (§2.3) of non-autoregressive translation models, Gu et al. (2018) showed positive results with sequence-level knowledge distillation (Kim and Rush, 2016). In general terms, this can be achieved by sampling a translation of the source sequences in the train set. When an autoregressive teacher model is available, one can achieve this by decoding the source sequence with the teacher model using the beam search algorithm as usual. The resulting translations constitute a new, distilled dataset. This kind of knowledge distillation makes the resulting training targets less noisy and more deterministic, ensuring that for instance "Thank you" will be consistently translated into the same German translation (§2.3).

Prior works from different disciplines provided theoretical support for the impact of distillation in translation. From a machine learning perspective, Zhou et al. (2020) showed distillation reduces the conditional entropy of the translations given the source sequences. They further showed distilled targets contained more words monotonically aligned with their direct translations in the source sequence. We argue this phenomenon can be viewed as *syntactic conditional entropy*, measuring the amount of uncertainty in the sentence structures. From a linguistic perspective, Bangalore et al. (2015) showed translations with low syntactic entropy are easier to produce.

As stated in §2.3, diffusion-based machine translation models do not follow the conditional independence assumption of NAT models, as they generate a trajectory of sequences $(\mathbf{x}^{(t)})_{1 \leq t \leq T}$ where for $1 \leq t < T, 1 \leq s \leq S$ the column $\mathbf{x}_i^{(t)}$ (which corresponds to a token embedding vector) is influenced by the whole sequence $\mathbf{x}^{(t+1)}$. Li et al. (2022) found that empirically learned word embeddings formed clusters of words with the same part-of-speech tags. Generally a diffusion models noise schedule should be rather smooth with no major jumps, so that the individual columns of the trajectory first drift towards a cluster of word embeddings early and commit to a single embedding later in the process. This leads to the assumption that a sequence's syntactic structure is first decided, before the model finally commits to individual words.

The syntactic conditional entropy of a training dataset could lead to a multimodality problem of diffusion models, where different syntactic structures represent the different modes in the early diffusion process. As distilled datasets exhibit lower syntactic conditional entropy, sequence-level knowledge distillation could improve the results of diffusion-based machine translation models. Motivated by this, we investigate how knowledge distil-

lation impacts the translation performance of diffusion models.

3.4 Autoregressive Sampling

Besides empirical successes of autoregressive models, the sequential nature of text suggests that generating one token at a time is a promising approach to text generation. Consequently, we wonder whether diffusion-based machine translation systems are limited in performance by fixing all tokens of the generated sequence at once. Indeed, a very recent work (Yuan et al., 2022) showed improvements by considering the sequential nature of the outputs, more specifically by learning to apply different noise levels to each token at every time step.

In the context of our approach, Diffusion-LM approaches the inclusion of prior data by the infilling algorithm. Building upon that, we propose an iterative sampling method, where in the each iteration i the first $i-1$ tokens of the last iterations output are served to the model as prior information. This conditional probability is approximated by the infilling algorithm.

So by sampling

$$(\mathbf{x}_1^{(i)},\ldots,\mathbf{x}_S^{(i)}) \sim p(\mathbf{x}_1,\ldots,\mathbf{x}_S | \mathbf{x}_1=\mathbf{x}_1^{(i-1)},\ldots,\mathbf{x}_{i-1}=\mathbf{x}_{i-1}^{(i-1)}) \tag{3a}$$

and discarding $(\mathbf{x}_{i+1}^{(i)},\ldots,\mathbf{x}_S^{(i)})$ we approximate

$$p(\mathbf{x}_i^{(i)} | \mathbf{x}_1=\mathbf{x}_1^{(1)},\ldots,\mathbf{x}_{i-1}=\mathbf{x}_{i-1}^{(i-1)}) \tag{3b}$$

yielding the usual autoregressive formula:

$$\hat{p}(\mathbf{x}_1^{(1)},\ldots,\mathbf{x}_S^{(S)}) = p(\mathbf{x}_1^{(1)}) \prod_{i=2}^{S} p(\mathbf{x}_i^{(i)} | \mathbf{x}_1=\mathbf{x}_1^{(1)},\ldots,\mathbf{x}_{i-1}=\mathbf{x}_{i-1}^{(i-1)}) \tag{3c}$$

Algorithm 1 describes the sampling algorithm in detail. For the naive implementation given there, this increases the time needed for decoding by a factor of $\mathcal{O}(S)$. However, when detecting the end of the generation process, this factor is in $\mathcal{O}$(average generated sequence length).

4 Experimental Setup

Dataset and Preprocessing We use the German-English text-to-text partition of the CoVoST (Wang et al., 2020) dataset and train the models for German-to-English translation. This dataset was chosen due to its comparable size to the experimental setup of Li et al. (2022), which used 50K to 98K samples in training. Due to the slow decoding process of the autoregressive sampling method, the test set was reduced to a subset of 2010 samples.[7] The dataset statistics are in Table 1. Details on preprocessing are in Appendix B.

Algorithm 1 Autoregressive sampling

Input
$s \in V^l, 1 \le l \le \frac{S}{2}$ The source sequence
▷ Initialize the translation as the empty word
$t \leftarrow \epsilon$
for $k = l+2, \ldots, S$ **do**
  $j \leftarrow s \oplus (\hat{s}) \oplus t$
  Pad j up to length S
  ▷ Embed concatenated sequence
  $\tilde{\mathbf{x}} \leftarrow E_\theta(j)$
  ▷ Calculate the mask
  $m_i = 1$ for $1 \le i \le |j|$
  $m_i = 0$ for $|j|+1 \le i \le S$
  Draw $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, I_{d \times S})$
  for $t = T-1, \ldots, 0$ **do**
    ▷ With Σ_{t+1} as described in equation (1a).
    Draw $\mathbf{x}_t \sim \mathcal{N}(\mu_\theta(\mathbf{x}_{t+1}, t+1), \Sigma_{t+1})$
    ▷ Overwrite where the data is given by $\tilde{\mathbf{x}}$
    $\mathbf{x}_{t;i} \leftarrow \tilde{\mathbf{x}}_i$ where $m_i = 1$
  end for
  ▷ Recover the most likely token for position k
  $t \leftarrow t \oplus \operatorname{argmax}_{j \in V} R_\theta(\mathbf{x}_0)_{k,j}$
end for
▷ Return the sequence of generated tokens t
return t

Split	# samples	Avg. source len.	Avg. target len.
train	127,638	12.6	12.41
valid	13,510	13.16	13.02
test (reduced)	2,010	13.6	13.47

Table 1: Key metrics of the dataset CoVoST, with the tokenizer used here and the reduced test split.

Evaluated Diffusion Models We evaluate 4 types of diffusion models described in §3:

1. **Diffusion-LM-MT**: Diffusion-LM adapted with classifier-free diffusion (§3.1)
2. **DiffuSeq**: the standard DiffuSeq model (§3.2)
3. **DiffuSeq, Distilled**: DiffuSeq with sequence-level knowledge distillation (§3.3)
4. **DiffuSeq, AR-Sampling**: DiffuSeq with autoregressive sampling (§3.4)

We use a max length of 64 tokens following Diffusion-LM (Li et al., 2022). For the knowledge distillation dataset, we use the pretrained model by Ng et al. (2019) as the teacher model. More details on the model architectures are in Appendix C. For all models, the encoder in the diffusion kernel is parameterized by a network following the BERT-based architecture. All weights are initialized randomly following Li et al. (2022).

[7] The reduced test set is available under https://drive.google.com/file/d/1nj2S7dOLGBel7ZR4AWbVCxEFVcgWg_V3/view?usp=drive_link

Model	BLEU↑	COMET↑
Diffusion-LM-MT	2.2	39.2
DiffuSeq	10.0	48.0
DiffuSeq, AR-Sampling	10.7	48.1
DiffuSeq-distilled	12.5	49.7
Transformer	**28.7**	**72.2**
Levensthein-Transformer	18.5	61.4

Table 2: Direct comparison of models by BLEU score and COMET score under the `wmt22-comet-da` score.

Sampling All models used a step size of 1 during the sampling process. This results in a very long decoding time, as the diffusion kernel needs to be evaluated in every iteration. Using a lower number of diffusion steps during sampling accelerates the sampling process, but generally leads to decreased performance (Li et al., 2022; Gong et al., 2023).

Baselines We use a Transformer model (Vaswani et al., 2017) as the main baseline. Given the non-autoregressive nature of diffusion models, we also compare to Levenshtein Transformer (Gu et al., 2019), an established NAT model. More details on the baselines are in Appendix D.

Evaluation The detokenized results of all implementations and baselines were evaluated by BLEU-scores by SacreBLEU (Post, 2018) and by the `wmt22-comet-da` model (Rei et al., 2022), which is the default COMET model at the time of writing. Both scores are reported as $\times 100$ for readability.

5 Results and Discussions

5.1 Translation Quality

The results of the proposed models and the baselines are presented in Table 2. All diffusion-based models heavily underperformed compared to both the Transformer model and the Levensthein-Transformer with a large gap of over 15 BLEU.

Compared to the standard DiffuSeq, the model employing sequence-level knowledge distillation (**DiffuSeq-distilled**) showed a unclear improvement of $+2.5$ BLEU and $+1.7$ COMET. This provides some support to our hypothesis in §3.3 on knowledge distillation's positive role in face of the multimodality problem. The model with autoregressive sampling method (**DiffuSeq, AR-Sampling**) brings a gain of $+0.7$ BLEU but does not improve the COMET score. Therefore, whether this approach has any impact on translation quality remains unclear. This suggests that the inclusion of prior knowledge by the infilling algorithm has little impact on the model's generation process. The **Diffusion-LM-MT** model, modeling the joint distribution performed poorly when faced with the task of translating the test data. When sampling from the Diffusion-LM model without the infilling algorithm, the model successfully generated pairs of German and English sentences. The data generated by this unguided approach, when evaluated by the reference-free COMET model `wmt20-comet-qe-da` (Rei et al., 2020) achieved a score[8] of 8.72. However, when faced with the challenge of translating the test set, the score fell to 0.94. This suggests that the infilling algorithm in its current form is ill fit to properly approximate conditional distributions as complex as machine translation tasks.[9]

5.2 Impact of Source Lengths

Next we investigate the impact of the input length on the translation quality of all models in the experiments. When the translated samples are split into buckets of roughly equal size by the length of the source sequence, we notice the diffusion language models fall off notably faster in BLEU score compared to the baseline transformer model, suggesting that *long-range dependencies* might be more problematic for these models to capture.

We formally test this by evaluating the relative difference in BLEU scores $d_r(\text{BLEU}_a, \text{BLEU}_b) := \frac{\text{BLEU}_a - \text{BLEU}_b}{\max\{\|\text{BLEU}_a\|, \|\text{BLEU}_b\|\}}$ between pairs of translation systems a and b. The relative difference followed linear trends, so we performed a t-test of slopes, testing against the null hypothesis "The relative difference in BLEU is uncorrelated to the length of the source sequence in tokens.". The resulting test statistics and the statistically significant results are in Table 3.

While the results from the autoregressive sampling method for DiffuSeq are slightly better than those of the standard sampling procedure for long source sequences, our experiments did not provide statistically significant data indicating that this method provides a particular benefit on long sequences. Furthermore, the DiffuSeq-distilled model utilizing knowledge distillation achieves

[8] Scores are not comparable to those in Table 2 due to a different COMET model with reference-free evaluation.

[9] This might indicate that during the generation process interdependencies within the German and English sentence are generally more influential than the cross dependency between the sequences, which also provides an explanation for the improved performance of the DiffuSeq model. This hypothesis would need further testing however.

Models	DiffuSeq	DiffuSeq, AR	Transformer	Diffusion-LM-MT	Lev-Transformer	DiffuSeq-distilled
DiffuSeq	–	0.40	**11.29**	-2.89	**4.95**	-0.20
DiffuSeq, AR	-0.40	–	**9.88**	-2.55	**4.06**	-0.56
Transformer	-11.29	-9.88	–	-10.42	-7.36	-10.97
Diffusion-LM-MT	**2.89**	2.55	**10.42**	–	**7.33**	**2.69**
Lev-Transformer	-4.95	-4.06	**7.36**	-7.33	–	-4.97
DiffuSeq-distilled	0.20	0.56	**10.97**	-2.69	**4.97**	–

Table 3: Test statistics for the t-test of slopes with critical value $t_{1987}(0.995) \simeq 2.58$ for a 1% significance level. Pairs where the null hypothesis "The relative difference of scores is uncorrelated to the length of the source sequence" can be rejected and where the slope is positive are marked in bold. By this, a positive test statistic indicates a significant impact of the length of the source sequence on the relative performance of the models, indicating that the model at the top of the column performs relatively better on longer sequences than the model at the start of the row.

Models	Training Time	# of steps	Batch Size	Decoding Time	GPUs
Lev-Transformer	19h	300,000	128	6s	1 NVIDIA RTX 3070
Diffusion-LM	3d 6h	600,000	128	1h 44m 46s	1 NVIDIA TITAN RTX
DiffuSeq	14d 9h	80,000	2048	3h 23m 17s	1 NVIDIA TITAN RTX
DiffuSeq AR	14d 9h	80,000	2048	>30h	1 NVIDIA TITAN RTX
DiffuSeq-distilled	10d 12h	60,000	2048	3h 23m	1 NVIDIA TITAN RTX

Table 4: Key metrics on the training and decoding times of the different non-autoregressive models. Decoding times are reported for the entire reduced test set containing 2010 samples. When re-evaluating the decoding time for the Levenshtein Transformer after the initial submission, times between 8.4 and 14.8 seconds were measured.[10]

higher scores than the standard DiffuSeq model overall, but follows the same trends as the standard DiffuSeq model over increasing length of the source sequences.

The non-autoregressive Levenshtein Transformer consistently outperforms all diffusion-based models, but also falls off faster than the autoregressive Transformer model on longer sequences.

5.3 Training and Decoding Time Comparison

Key metrics on the training and decoding times of the various non-autoregressive models are summarized in Table 4. The diffusion-based models suffer from long training and decoding times. For training time, the slow optimization process can be explained by two factors. Firstly, the model essentially faces the problem of guessing the whole target sequence based on the source sequence by a single evaluation of an encoder stack, which is a very hard problem. Secondly, the DiffuSeq and DiffuSeq-distilled models both rely on large batch sizes to avoid converging to trivial distributions. The slow decoding speed on the other hand is largely explained by the number of diffusion steps, as the decoding process in our case requires 2000 iterations of the encoder stack. A remedy would be to down-sample the number of diffusion steps taken (Song et al., 2021) at the cost of sample quality (Gong et al., 2023).

5.4 Translation Samples

Some translation examples by the different models are shown in Table 5. With the shortest input, all systems are able to translate correctly apart from the Diffusion-LM-based models. With the two longer input sequences, despite mostly capturing the rough meaning of the input, the non-autoregressive models in general exhibit problems with output fluency. An exception is the Diffusion-LM-based model which hallucinates translations that are unrelated to the input. This is an indication that the conditional information from the source is disregarded by the model.

5.5 Open Questions

Tackling Multimodality The experiment results indicate that similarly to other NAR models the multimodality problem presents a challenge to diffusion models, with knowledge distillation providing clear benefits for the performance and convergence properties of diffusion-based MT models. The improved performance of the model utilizing sequence-level knowledge distillation is likely due to decreased nondeterminism in the dataset, which is in-line with other findings on non-autoregressive translation systems (Gu et al., 2018). Consequently, the applicability of other methods employed to tackle the multimodality problem in NAR models

[10]The DiffuSeq-based models still showed improvement even after extensive training duration

Source	Robert Simonds ist verheiratet.
Target	Robert Simonds is married.
Transformer	Robert Simonds is married.
Diffusion-LM	Robert -ieew is married.
DiffuSeq	Robert Simonds is married.
DiffuSeq, AR	Robert Simonds is married.
Lev-Transformer	Robert Simonds is married.
DiffuSeq-distilled	Robert Simonds is married.
Source	Der Duft von Fruehling stroemte in ihre Nase.
Target	The fragrance of spring floated into her nose.
Transformer	The The fragrance of spring running in her nose.
Diffusion-LM	The of of them ran about the body in the basement.
DiffuSeq	The frag of remain in their nose.
DiffuSeq, AR	The frag of internationally ended in their nose..
Lev-Transformer	The jce of spring, and comes in their ne.
DiffuSeq-distilled	The frag of their fragrance in their nose.
Source	Gleichzeitig wurde mit der Elektrifizierung des Netzes begonnen.
Target	Electrification of the network began at the same time.
Transformer	At the same time, the electrification of the network was started.
Diffusion-LM	It was closed with the populationun of fin during city of Baden.
DiffuSeq	At the same time , the similar railway board has been areas in the network.
DiffuSeq, AR	At the same time , the upper sub sh of the network was moved.
Lev-Transformer	At the same time, the electrification was started with the netnetwork.
DiffuSeq-distilled	At the same time , Soviet inv independent of the estate was started.

Table 5: Examples of translations from the different systems.

to diffusion-based MT models should be studied.

Output Diversity A potential advantage of diffusion models is the diversity of the generated outputs. We did not explore how knowledge distillation affects the diversity score of the system. Gong et al. (2023) showed that DiffuSeq scores high in the diverse 4-gram (Deshpande et al., 2019) score measuring the ration of distinct 4-grams in a set of outputs for one source sequence. Quite possibly the increased quality of samples when using sequence-level knowledge distillation comes at a trade-off for decreased diversity of generation outputs. The diversity of results given different seeds for the generation process can also be leveraged by applying Minimum Bayes Risk (MBR) decoding (Kumar and Byrne, 2004), where each candidate from a set of translations is assigned a risk based on how similar it is to the other candidates. The candidate with the lowest risk is then chosen as the system output (Li et al., 2022; Gong et al., 2023).

Further Improving Diffusion-LM-MT The experiment results show the Diffusion-LM-based model performed poorly while the standard DiffuSeq achieved acceptable scores. The generation process of DiffuSeq is also equivalent in implementation to the infilling procedure utilized by Diffusion-LM and the autoregressive sampling method, but unlike the infilling procedure used there, with DiffuSeq the conditioning information is served to the model in the same way during training (Gong et al., 2023). This could motivate a hybrid AR/NAR approach, using a diffusion-based system to generate few tokens at a time.[11]

6 Conclusion

Using sequence-level knowledge distillation we saw a clear improvement in both training speed and model performance of diffusion-based machine translation systems. We believe they benefit from the reduced syntactic conditional entropy of distilled datasets and conclude that they suffer of a form of the multimodality problem, similarly to other NAR machine translation systems. Based on this, methods employed in other NAR models to help them handle multimodality in the data are likely to improve the performance of diffusion-based machine translation approaches.

The Diffusion-LM-MT model proved capable of expressing the joint density of source and translation implying that with an improved infilling algorithm good conditional densities could be sampled from these models. However, using the infilling algorithm, it was ill-fit to produce high quality samples when used for a sequence to sequence task in our experiments. In a similar manner, the method of autoregressive sampling for diffusion-

[11] Initial experiments often converged to trivial distributions. To this end, the model should be refined to allow for sequences of variable length.

based translation systems had little impact on the quality of samples.

The diffusion-based MT models studied currently struggle with training and inference speed. While some factors contributing to the slow optimization of these models such as the static sequence length may be alleviated, the problem of predicting the initial sequence based of the noisy version remains difficult. Inference speed on the other hand can be improved with methods such as DDIM (Song et al., 2021) and newer work on Diffusion Models for the image domain likely could be applied to Diffusion Language Models as well. When using the results of Diffusion Language models directly without using MBR decoding, the models still fall decidedly behind the Transformer-based baselines. At the same time, MBR-decoding does not seem broadly applicable, as long as inference is still as slow as in current models.

Limitations

Comparison to SOTA translation models In the comparison to diffusion-based models, our Transformer model was a Transformer-base and was trained on a small dataset with around 100K parallel sentences. For a comparison to state-of-the-art translation models, one should use a larger model trained on over millions of sentence pairs, potentially initialized from pretrained weights. Therefore, the gap between diffusion-based models and state-of-the-art translation models is likely even larger than reported in this paper.

Decoding speed In the current form, the experimented diffusion-based models are prohibitively slow. Even when the quality gap to standard translation models is closed, the decoding speed renders these models unrealistic for deployed systems.

Acknowledgement

We thank the anonymous reviewers for insightful feedback. Part of this work was supported by funding from the pilot program Core-Informatics of the Helmholtz Association (HGF).

References

Srinivas Bangalore, Bergljot Behrens, Michael Carl, Maheshwar Gankhot, Arndt Heilmann, Jean Nitzke, Moritz Schaeffer, and Annegret Sturm. 2015. The role of syntactic variation in translation and post-editing. *Translation Spaces*, 4(1):119–144.

Jiaao Chen, Aston Zhang, Mu Li, Alex Smola, and Diyi Yang. 2023. A cheaper and better diffusion language model with soft-masked noise. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4765–4775, Singapore. Association for Computational Linguistics.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. 2023. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems XIX, Daegu, Republic of Korea, July 10-14, 2023*.

Aditya Deshpande, Jyoti Aneja, Liwei Wang, Alexander Schwing, and D. A. Forsyth. 2019. Fast, diverse and accurate image captioning guided by part-of-speech.

Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and Lingpeng Kong. 2023. DiffuSeq: Sequence to sequence text generation with diffusion models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor O. K. Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Jiatao Gu and Xiang Kong. 2021. Fully non-autoregressive neural machine translation: Tricks of the trade. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 120–133, Online. Association for Computational Linguistics.

Jiatao Gu, Changhan Wang, and Junbo Zhao. 2019. Levenshtein transformer. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 11179–11189.

Jindřich Helcl, Barry Haddow, and Alexandra Birch. 2022. Non-autoregressive machine translation: It's not as fast as it seems. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1780–1790, Seattle, United States. Association for Computational Linguistics.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Jonathan Ho and Tim Salimans. 2022. Classifier-free diffusion guidance. *CoRR*, abs/2207.12598.

Myeonghun Jeong, Hyeongju Kim, Sung Jun Cheon, Byoung Jin Choi, and Nam Soo Kim. 2021. Diff-TTS: A Denoising Diffusion Model for Text-to-Speech. In *Proc. Interspeech 2021*, pages 3605–3609.

Jungo Kasai, Nikolaos Pappas, Hao Peng, James Cross, and Noah A. Smith. 2021. Deep encoder, shallow decoder: Reevaluating non-autoregressive machine translation. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.

Diederik P. Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. 2021. Variational diffusion models. *CoRR*, abs/2107.00630.

Shankar Kumar and William Byrne. 2004. Minimum Bayes-risk decoding for statistical machine translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 169–176, Boston, Massachusetts, USA. Association for Computational Linguistics.

Xiang Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B. Hashimoto. 2022. Diffusion-lm improves controllable text generation. In *NeurIPS*.

Jindřich Libovický and Jindřich Helcl. 2018. End-to-end non-autoregressive neural machine translation with connectionist temporal classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3016–3021, Brussels, Belgium. Association for Computational Linguistics.

Andreas Lugmayr, Martin Danelljan, Andrés Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. 2022. Repaint: Inpainting using denoising diffusion probabilistic models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 11451–11461. IEEE.

Calvin Luo. 2022. Understanding diffusion models: A unified perspective. *CoRR*, abs/2208.11970.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook FAIR's WMT19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 314–319, Florence, Italy. Association for Computational Linguistics.

Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. 2022. GLIDE: towards photorealistic image generation and editing with text-guided diffusion models. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 16784–16804. PMLR.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Bo Qin, Aixin Jia, Qiang Wang, Jianning Lu, Shuqin Pan, Haibo Wang, and Ming Chen. 2022. The RoyalFlush system for the WMT 2022 efficiency task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 671–676, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Ricardo Rei, José G. C. de Souza, Duarte Alves, Chrysoula Zerva, Ana C Farinha, Taisiya Glushkova, Alon Lavie, Luisa Coheur, and André F. T. Martins. 2022. COMET-22: Unbabel-IST 2022 submission for the metrics shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 578–585, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. Unbabel's participation in the WMT20 metrics shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 911–920, Online. Association for Computational Linguistics.

Machel Reid, Vincent J. Hellendoorn, and Graham Neubig. 2022. Diffuser: Discrete diffusion via edit-based reconstruction. *CoRR*, abs/2210.16886.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 2256–2265. JMLR.org.

Jiaming Song, Chenlin Meng, and Stefano Ermon. 2021. Denoising diffusion implicit models. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Changhan Wang, Juan Pino, Anne Wu, and Jiatao Gu. 2020. CoVoST: A diverse multilingual speech-to-text translation corpus. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 4197–4203, Marseille, France. European Language Resources Association.

Hongyi Yuan, Zheng Yuan, Chuanqi Tan, Fei Huang, and Songfang Huang. 2022. Seqdiffuseq: Text diffusion with encoder-decoder transformers. *CoRR*, abs/2212.10325.

Chunting Zhou, Jiatao Gu, and Graham Neubig. 2020. Understanding knowledge distillation in non-autoregressive machine translation. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Hao Zou, Zae Myung Kim, and Dongyeop Kang. 2023. A survey of diffusion models in natural language processing.

A Additional Visualization

An example of the forward process is shown in Figure 2. Here, the initial distribution was a mixture distribution of two normal distributions, seen at $t = 0$. Over the course of the forward process noise was added, resulting in the density curve for $t = 2000$ resembling a standard normal distribution.

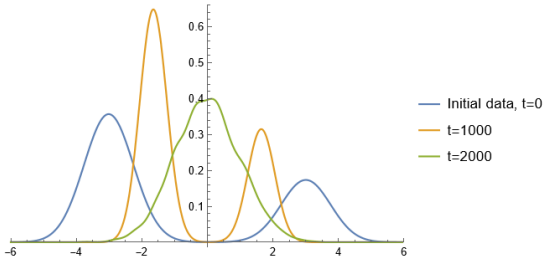

Figure 2: Smooth histograms of sampled values at X_0, X_{1000}, X_{2000} where $T = 2000$, based on 2000 simulations.

B Details on Preprocessing

For preprocessing, the special characters ä, ü, ö, and ß were replaced by ae, ue, oe and ss respectively, accents were removed, and the set of characters was reduced to the alphabet, numerals, and punctuation marks (excluding brackets and parentheses).[12] The texts are tokenized by Byte-Pair Encoding (BPE) (Sennrich et al., 2016) with a vocabulary size of 30,000. After tokenization, the training data for the diffusion-based models are further filtered by removing sequences longer than 64 tokens. This accounted for less than 0.01% of the samples in the dataset.

C Details on Model Architectures

1. **Diffusion-LM-MT**: This is the model based on Diffusion-LM with infilling as described in Section 3.1. Model hyperparameters:
   (a) **Embedding dimension: 256**
   (b) Diffusion steps: 4000
   (c) Noise schedule: "sqrt"
   (d) Estimated mean parameterized by estimating x_0
   (e) Batch size: 128
   (f) Maximum sequence length: 64
   (g) No gradient clipping
   (h) Fixed noise schedule
   (i) End-to-end training of diffusion kernel and embedding matrix
   (j) Decoding with the clamping trick applied[13]

2. **DiffuSeq**: The standard DiffuSeq model. Model hyperparameters are as given by Gong et al. (2023).

3. **DiffuSeq, AR**: The standard DiffuSeq model with the method of autoregressive sampling as described in Section 3.4.

4. **DiffuSeq-distilled**: The model utilizing sequence-level knowledge distillation as described in Section 3.3, using the same hyperparameters. We used the `wmt19-de-en` model by Ng et al. (2019) as the teacher model.

[12] This preprocessing was motivated by the idea of potentially leveraging closely related vocabulary between German and English and reducing the vocabulary size. The deviation from standard translation preprocessing steps (removing brackets and parentheses) could slightly affect the compatibility to other systems.

[13] The paper introducing Diffusion-LM states that this empirically improves sample quality (Li et al., 2022), however, some more recent papers suggest that this might not consistently be the case (Yuan et al., 2022)

D Details on Baselines

Transformer The model is with 6 layers, embedding dimension 512, feed-forward layer embedding dimension 1024 and 4 attention heads in both encoder and decoder. The model uses shared weights for encoder and decoder embeddings and for the language modeling head. Besides these parameters, the other parameters are the same as the original paper (Vaswani et al., 2017). Decoding was performed with beam size of 10, length penalty of 1, temperature of 1, and no further modifications to the standard beam search.

Levenshtein Transformer We follow the implementation here. Decoding parameters were also chosen as presented by the paper.