Nearest-neighbor-assisted Fine-tuning for Neural Machine Translation

First Author
Affiliation / Address line 1
Affiliation / Address line 2
Affiliation / Address line 3
email@domain

Second Author
Affiliation / Address line 1
Affiliation / Address line 2
Affiliation / Address line 3
email@domain

Abstract

Recent advancements in nearest-neighbor machine translation (kNN-MT) have showcased its potential to enhance domain adaptation in machine translation without the need for fine-tuning. Our initial exploration revealed a significant boost in kNN-MT's overall efficacy when applied to a fine-tuned MT model, with the kNN distributions more closely aligning with the ground truth targets. This observation prompts a crucial research question: *how can we best fuse the kNN methodology with fine-tuning to amplify domain adaptation results*? In response, we introduce *trainable-k*NN-MT, a novel approach that leverages statistics from kNN predictions to dynamically adjust gradients during fine-tuning. Compared to conventional fine-tuning, we report consistent improvements for established domain-specific translation tasks by as much as 1.39, 1.90, and 1.31 SacreBLEU and 1.03, 1.34, and 1.21 chrF for German-English, English-German, and English-Chinese translations, respectively. Furthermore, the *trainable-k*NN-MT not only helps mitigate overfitting concerns, but also enhances the effectiveness of the kNN-MT algorithm when the kNN predictions are interpolated during inference.

1 Introduction

Domain adaptation plays a crucial role in domain-specific machine translation. A generalized machine translation (MT) model may not effectively capture the subtle nuances of distinct domains. By adapting a machine translation system to a specific domain, one can enhance its accuracy, fluency, and overall translation efficiency (Chu and Wang, 2018; Saunders, 2022). Among the various domain adaptation approaches for machine translation, fine-tuning is pivotal, since it leverages the knowledge from a pre-trained MT model developed on extensive, general datasets, and specializes it for a new domain, allowing for a rapid adaptation with potentially fewer data (Chu et al., 2017). In addition, the k-nearest-neighbor machine translation (kNN-MT) (Khandelwal et al., 2021) has seen recent successes for domain adaption in machine translation, enhancing German-English domain-specific translations by a significant 9.2 BLEU over the base MT model without training on these domains.

While both fine-tuning and the kNN-MT method offer advantages, they also come with their limitations. Specifically, fine-tuning can be prone to overfitting (Kirkpatrick et al., 2017; Barone et al., 2017)—a scenario where a model is too closely tailored to a particular datasets and loses the capacity to generalize well to new, unseen data. On the other hand, the kNN-MT method faces a domain mismatch. The retrieval datastore, constructed based on the MT model weights developed on out-of-domain data, does not often align well with the intended target domain, thereby undermining the kNN model's efficacy (Cao et al., 2023).

In our initial explorations, we observed a notable improvement in the overall performance of the kNN-MT algorithm when we implemented the kNN retrieval based on a fine-tuned MT model, as depicted in Figure 1. In addition, with improved hidden states, the kNN prediction distributions aligned more closely with the ground truth targets, as evidenced in Table 6 (See in Appendix A). It seems that the statistical patterns within the kNN predictions might offer a new perspective on model fine-tuning extent. These findings prompt a potential research question: *what is the best strategy to combine the kNN-MT method with fine-tuning to further elevate the domain adaptation performance?* While the recent literature has seen a growing interest in nearest neighbor models, the primary focus has been on refining the efficiency and efficacy of the kNN search (Zheng et al., 2021; Wang et al., 2022; Jiang et al., 2022). Only a handful of studies, such as (Yang et al., 2022), have delved into leveraging the kNN method to bolster model fine-tuning for superior domain adaptation outcomes.

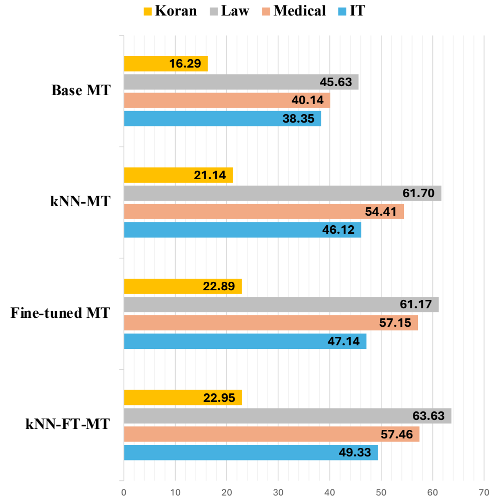

Figure 1: Performance of various models on German-English domain-specific translations, including Base MT, kNN-MT, Fine-tuned MT, and the kNN algorithm applied to the Fine-tuned MT, referred to as kNN-FT-MT. The metric used is SacreBLEU (Post, 2018).

In this paper, we propose *trainable-k*NN-MT, a novel method that combines fine-tuning with the kNN model. It basically learns translations conditioned on the statistical information extracted from the retrieved k-nearest-neighbors. We hypothesize that an optimally trained neural machine translation (NMT) model would enable the kNN search to retrieve the ground truth target as the most probable in the kNN distribution. Any deviation from this behavior might suggest potential for further refining the NMT model's learning. To address this, we design an approach which utilizes the ground-truth target probability within the kNN distribution to integrate the statistic patterns within the kNN predictions into model fine-tuning, which dynamically modulate gradients for back-propagation. This dynamic gradient adjustments aim to help the model intensify its learning in areas where its proficiency is lacking, while moderating its focus when the model already demonstrates adeptness in translating certain in-domain texts. In addition, it is important that, as the model undergoes fine-tuning, the datastore, constructed on in-domain data using the NMT model weights, evolves in tandem with the model's progression, which is updated with the latest model weights after each training epoch.

In our experiments, the *trainable-k*NN-MT not only surpasses the conventional fine-tuning, establishing itself as an innovative fine-tuning approach for neural machine translation, but also helps mitigate overfitting concerns, demonstrating superior generalization to unseen domains. Furthermore, it enhances the effectiveness of the kNN-MT algorithm when interpolating the kNN retrieval during inference.

2 Related Work

There has been a recent surge in interest in the non-parametric kNN-MT. Most studies have focused on improving the efficiency and effectiveness of this method. For instance, Zheng (2021) introduced a Meta-k network to dynamically retrieve candidates, which utilized neighbor importance to improve the translation performance. Additionally, Wang (2022) suggested a cluster-based pruning solution to filter out redundancies in the datastore for kNN search to improve inference efficiency. Moreover, Jiang (2022) analyzed the impact of noise from the kNN retrieved set and proposed a confidence-enhanced kNN-MT model to improve its robustness. However, only a few studies have utilized the kNN method to enhance model training. For example, Yang (2022) utilized a knowledge distillation framework to enhance the NMT performance, by distilling the kNN search as a teacher model to guide the NMT training.

3 Methodology

In this section, we will present the *trainable-k*NN-MT, which leverages the statistical patterns within kNN predictions to enhance model fine-tuning through dynamic gradient adjustments.

3.1 Preliminary Method

In a typical neural machine translation setup, given a source sentence x and the target prefix tokens $y_{1:i-1}$, the model predicts the next target token y_i with as $P_{\text{MT}}(y_i|x, y_{1:i-1})$, sourced from a softmax distribution over the vocabulary.

In kNN-MT (Khandelwal et al., 2021), for a sequence of source tokens and a sequence of target prefix tokens $(s, t_{1:i-1})$ from the in-domain data $\mathcal{D}$, the pre-trained NMT model produces the hidden state $f(s, t_{1:i-1})$ of the i-th target token t_i to construct a datastore. The definition of the datastore is as follows:

$$(\mathcal{K}, \mathcal{V}) = \\ \{(f(s, t_{1:i-1}), t_i), \forall t_i \in t \mid (s, t) \in \mathcal{D}\},$$

where $\mathcal{K}$ represents all keys, while $\mathcal{V}$ represents all corresponding values.

During inference, given the hidden state of a translation context as a query, the kNN-MT will first retrieve top-k nearest neighbors from the above datastore. The retrieved set is then converted into the distribution $P_{\text{kNN}}(y_i|x, \hat{y}_{1:i-1})$ over the vocabulary by,

$$P_{\text{kNN}}(y_i|x, \hat{y}_{1:i-1}) \propto \\ \sum_{(k_j, v_j) \in \mathcal{N}} \mathbb{1}_{y_i = v_j} \exp(\frac{-d(k_j, f(x, \hat{y}_{1:i-1}))}{T}), \tag{1}$$

where $\mathcal{N}$ is the set of k nearest neighbors, and k_j, v_j are the key and value of the retrieved neighbors. T represents the temperature, flattening the distribution to prevent over-fitting to the most similar retrievals (Khandelwal et al., 2021).

Finally, the prediction of the next token y_i relies on the interpolation of the predictions from the NMT model and the kNN search as follows,

$$P_{\text{comb}}(y_i|x, \hat{y}_{1:i-1}) = \\ \lambda P_{\text{kNN}}(y_i|x, \hat{y}_{1:i-1}) + (1-\lambda) P_{\text{MT}}(y_i|x, \hat{y}_{1:i-1}), \tag{2}$$

where λ is a hyper-parameter for merging the two different distributions.

3.2 The *trainable*-kNN-MT

The structure of *trainable*-kNN-MT can be seen in Figure 2, featuring two main modules: datastore construction (left) and model fine-tuning (right).

Given a pre-trained NMT model, we initially construct a datastore on in-domain data based on the current NMT model weights. As we start fine-tuning using the in-domain data, for a specific source sentence x and the ground-truth target prefix tokens $y_{1:i-1}$ of a training instance, we first exclude the key-value pair associated with this translation context from the datastore. Afterwards, the kNN search retrieves the top-k nearest neighbors from the remaining key-value pairs in the datastore. The retrieved candidate set is subsequently converted into a distribution via Equation 1. Finally, statistics of this distribution are then utilized to adjust the gradient during model backpropagation.

The rationale for the initial exclusion is straightforward: without it, the kNN search might simply retrieve the key-value pair corresponding to the query itself, which would be self-referential and counterproductive.

Throughout the fine-tuning process, the datastore, constructed on in-domain data using the NMT model weights, is updated alongside the fine-tuning procedure. After each training epoch, the datastore is re-constructed based on the model's latest weights.

Originally, fine-tuning an NMT model optimizes the model parameters θ by minimizing the cross entropy loss on the in-domain training dataset $\mathcal{D}$ as follows,

$$\mathcal{L}_{ce} = \frac{1}{|\mathcal{D}|} \sum_{(x,y) \in \mathcal{D}} -\log P_{\text{MT}}(y_i|x, y_{1,i-1}; \theta). \tag{3}$$

For the *trainable*-kNN-MT, we leverage the probability of ground truth target within the kNN distribution to generate gradient adjustments for the NMT training. Based on our preliminary explorations, this probability might suggest how the NMT model's learning could be adjusted in certain contexts.

Given the source sentence x and ground truth prefix targets $y_{1:i-1}$, $P_{\text{kNN}}(y_i|x, y_{1:i-1})$ is a function of current NMT model weights and is differentiable during back propagation as follows,

$$P_{\text{kNN}}(y_i|x, y_{1:i-1}; \theta) \propto \\ \sum_{(k_j, v_j) \in \mathcal{N}} \mathbb{1}_{y_i = v_j} \exp(\frac{-d(k_j, f_\theta(x, y_{1:i-1}))}{T}), \tag{4}$$

where θ represents the NMT model weights, and k_j, v_j are the key and value of the retrieved neighbor from the datastore constructed from last epoch.

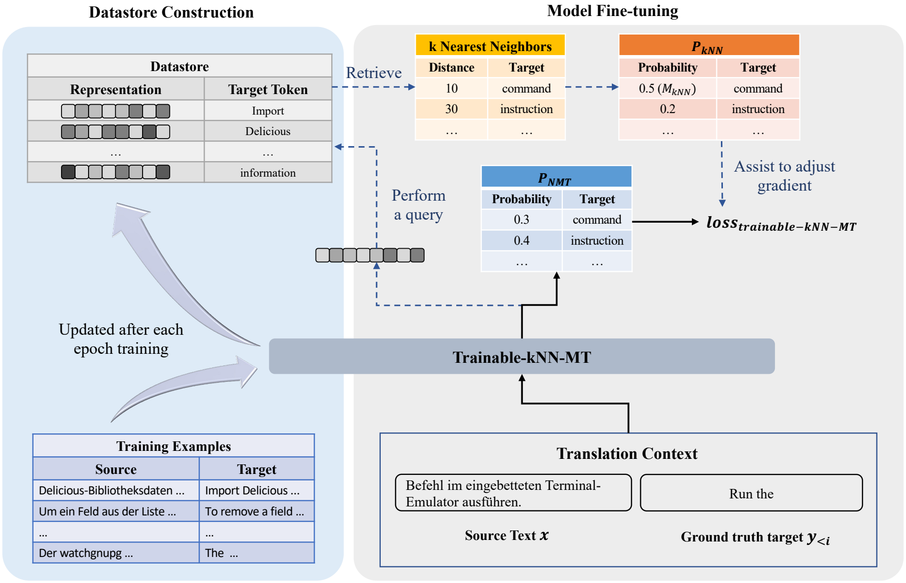

Figure 2: Schematic representation of the *trainable-k*NN-MT. Dashed lines indicate where translations are learnt with the assistance of statistics from kNN predictions. The datastore is updated after each training epoch with the latest NMT model weights.

We define a function $g_{k\text{NN}}$ as the probability associated with the ground truth y_i in the kNN distribution, expressed as follows:

$$g_{k\text{NN}} = P_{k\text{NN}}(y_i|x, y_{1:i-1}; \theta). \tag{5}$$

and the training loss is formulated as follows,

$$\mathcal{L} = \frac{1}{|\mathcal{D}|} \sum_{(x,y)\in\mathcal{D}} -\log g_{k\text{NN}} P_{\text{MT}}(y_i|x, y_{1,i-1};\theta) \tag{6}$$

Since $g_{k\text{NN}}$ is in $(0, 1)$, a lower value suggests the need to bolster the NMT model's learning in those particular contexts, leading to a greater gradient. Conversely, a higher value will prompt a relatively tempered gradient adjustment.

When $P_{k\text{NN}}(y_i|x, y_{1:i-1})$ is zero, it indicates that the ground truth target was not retrieved by the kNN search—a scenario meriting our attention. This outcome could be the result of an absence of similar contexts in the training data (e.g. rare words) or a lapse in the kNN search algorithm. Given such circumstances, we establish a baseline value for $g_{k\text{NN}}$. Considering the most extreme kNN distribution would be a uniform one where every prediction is equally probable with a likelihood of $1/k$, it's logical to assign $g_{k\text{NN}}$ a small value such as $1/k$. Through this adjustment, the trainable-kNN-MT enhances the NMT model's learning trajectory by amplifying the gradients used for updating the NMT model weights.

Inspired by Barone (2017), an alternative interpretation of Equation 6 suggests the incorporation of regularization into the NMT model. To elucidate this concept, let us consider the expanded form of Equation 6 as follows,

$$\mathcal{L} = -\frac{1}{|\mathcal{D}|} \sum_{(x,y)\in\mathcal{D}} \log g_{k\text{NN}} + \log P_{\text{MT}}(y_i|x, y_{1,i-1};\theta). \tag{7}$$

Since the datastore used in Equation 4 and 5 is constructed based on the model weights from the last epoch, the term $\log g_{k\text{NN}}$ in Equation 7 aims to enhance the model's generalization capabilities by penalizing the current in-domain model weights by their distance from the weights trained from last epoch via the kNN prediction. In this work, we investigate if $\log g_{k\text{NN}}$ may work as an regularization to potentially improve domain adaption performance while mitigating overfitting.

4 Experiments

In this section, we will present and discuss our experimental results in domain-specific translation tasks, specifically focusing on German-English, English-German, and English-Chinese translations.

4.1 Data

We adopt the experimental setup from kNN-MT (Khandelwal et al., 2021) for German-English domain-specific translations, using datasets from IT, Medical, Law, and Koran domains (Aharoni and Goldberg, 2020). We also reverse the source

Model	IT	Medical	Law	Koran	Avg.
Base MT	38.35 / 56.92	40.14 / 59.03	45.63 / 64.27	16.29 / 38.12	35.10 / 54.59
Fine-tuned MT	47.14 / 63.44	57.15 / 69.89	61.17 / 74.56	22.89 / 43.31	47.09 / 62.80
kNN-KD (Yang et al., 2022)	—	56.50 / —	61.89 / —	**24.86** / —	—
Base *trainable-k*NN-MT (Ours)	47.77 / 63.95	57.62 / 70.25	62.98 / 76.09	22.87 / 43.22	47.81 / 63.38
FT *trainable-k*NN-MT (Ours)	**49.31 / 65.04**	**58.28 / 70.71**	**63.41 / 76.27**	22.90 / 43.31	**48.48 / 63.83**

Table 1: Performance of the *trainable-k*NN-MT using **vanilla inference** without incorporating the kNN retrieval, on **German-English** domain-specific test sets. SacreBLEU / chrF scores are reported and averaged across domains for a comprehensive comparison. When compared with Fine-tuned MT, the *trainable-k*NN-MT enhances the overall performance by up to 1.39 BLEU and 1.03 chrF points.

Model	IT	Medical	Law	Koran	Avg.
Base MT	29.74 / 50.41	35.56 / 56.55	40.85 / 61.91	13.97 / 36.32	30.03 / 51.30
Fine-tuned MT	39.70 / 60.18	52.50 / 67.66	57.16 / 72.80	29.79 / 48.85	44.79 / 62.37
Base *trainable-k*NN-MT (Ours)	41.17 / 61.28	53.03 / 68.37	57.54 / 73.17	31.15 / 49.70	45.72 / 63.13
FT *trainable-k*NN-MT (Ours)	**41.65 / 61.49**	**54.09 / 68.80**	**58.29 / 73.67**	**32.74 / 50.88**	**46.69 / 63.71**

Table 2: Performance of the *trainable-k*NN-MT for **vanilla inference** on **English-German** domain-specific test sets, measured in SacreBLEU / chrF scores. Compared with Fine-tuned MT, the *trainable-k*NN-MT boosts the overall fine-tuning performance, achieving gains of up to 1.90 BLEU and 1.34 chrF points.

and target for English-German translations. For English-Chinese, we pre-train a base NMT model on the CCMT 2022 Corpus with 8.2 million pairs[1], then fine-tune using the Subtitle, News, and Thesis datasets (Tian et al., 2014)[2]. We pre-process the data by maximum length filtering with 250 to ensure data quality. For the English-Chinese datasets, English texts are tokenized using the Moses tokenizer, while Chinese texts undergo tokenization with Jieba. We then train a joint source-target BPE model (Sennrich et al., 2016) with 32,000 merge operations. This results in the construction of a joint vocabulary, comprising 48,672 subwords. Detailed statistics for the in-domain datasets can be found in Tables 7 and 8 within Appendix A.

4.2 Experimental Setup

Model Structure For the German-English and English-German translations, we adopt the winning systems from the WMT 2019 news translation tasks (Ng et al., 2019) as our base NMT models, which employ the Transformer big model architecture (Vaswani et al., 2017). For the English-Chinese translations, we initialize our experiments with a Transformer base model (Vaswani et al., 2017), pre-trained on the CCMT 2022 corpus.

Model Setting Beginning with a based NMT model pre-trained on out-of-domain data, we apply the *trainable-k*NN-MT for fine-tuning utilizing domain-specific data. Additionally, even for fine-tuned NMT models that have undergone prior conventional fine-tuning, employing the *trainable-k*NN-MT approach presents an opportunity for additional optimization and potential enhancement.

For initial fine-tuning phases, the duration of the *trainable-k*NN-MT process is set to match the epochs used for conventional fine-tuning approaches. Specifically, for German-English translations, we run 55, 60, 55, and 58 epochs respectively for the IT, Medical, Law, and Koran domains. For English-German translations, the epochs span 78, 74, 71, and 263 for the same domains. In the case of English-Chinese translations, the respective epoch counts for Subtitles, News, and Thesis domains are 182, 174, and 94.

When advancing to further fine-tuning of models already subject to optimization with *trainable-k*NN-MT, we adhere to a predefined duration of 30 epochs, consistent with the patience parameter from our standard fine-tuning practices. Across both stages, we employ the *trainable-k*NN-MT for gradient adjustment, ensuring a comprehensive

[1] Available for download on the WMT 2022 website: https://www.statmt.org/wmt22/translation-task.html.

[2] For each domain, there are only training and test sets. We randomly sample 2000 parallel sentence pairs from the training set to get the validation set.

Model	Subtitles	News	Thesis	Avg.
Base MT	24.36 / 20.31	30.43 / 26.79	25.15 / 22.22	26.65 / 23.11
Fine-tuned MT	33.26 / 32.75	34.58 / 30.10	47.52 / 43.76	38.45 / 35.54
Base *trainable-k*NN-MT (Ours)	34.06 / 34.37	34.41 / 29.98	49.22 / 45.59	39.23 / 36.65
FT *trainable-k*NN-MT (Ours)	**34.72 / 33.91**	**34.88 / 30.45**	**49.67 / 45.89**	**39.76 / 36.75**

Table 3: Performance of the *trainable-k*NN-MT for **vanilla inference** on **English-Chinese** domain-specific test sets, measured in SacreBLEU / chrF scores. Compared with Fine-tuned MT, the *trainable-k*NN-MT boosts the overall fine-tuning performance, achieving gains of up to 1.31 BLEU and 1.21 chrF points.

Model	IT	Medical	Law	Koran	Avg.
kNN-MT (Khandelwal et al., 2021)	46.12	54.41	61.70	21.14	45.84
Adaptive kNN-MT (Zheng et al., 2021)	47.20	55.71	62.64	19.39	46.24
CKMT (Wang et al., 2022)	47.94	56.92	62.98	19.92	46.94
Robust-kNN-MT (Jiang et al., 2022)	48.90	57.28	64.07	20.71	47.74
kNN-FT-MT	49.33	57.46	63.63	22.95	48.34
Base *trainable-k*NN-MT (Ours)	49.05	57.45	64.68	23.59	48.69
FT *trainable-k*NN-MT (Ours)	**49.69**	**58.39**	**64.78**	**23.84**	**49.18**

Table 4: Performance of the *trainable-k*NN-MT for ***k*NN inference** in SacreBLEU on **German-English** domain-specific test sets, integrating the kNN retrieval during inference. The FT *trainable-k*NN-MT outperforms baseline systems, gaining roughly 0.84 BLEU point over the kNN-FT-MT.

and tailored approach to model refinement.

At the training stage, the *trainable-k*NN-MT is initialized with a pre-trained NMT model and fine-tuned with the Adam algorithm (Kingma and Ba, 2015). The hyper-parameters of the kNN search are tuned on the validation sets beforehand, which are shown in Table 9, 10 and 11 in Appendix A. Notations of the models are described in the next section. We use a learning rate of 5e-04 for fine-tuning the base NMT model and 7e-05 for continuous fine-tuning the model already tuned with the conventional method. Experiments employ the Fairseq toolkit (Ott et al., 2019) on a single Tesla V-100 GPU with a 2048-token batch size and a 32-batch gradient accumulation. After each training epoch, the datastore is re-constructed with the latest NMT weights until convergence.

Evaluation Evaluation is conducted using SacreBLEU[3] and chrF metrics (Post, 2018; Popović, 2015)[4]. In our reports, (1) **Base MT** represents the base NMT model, which is pre-trained with out-of-domain data; (2) **Fine-tuned MT** represents the conventional fine-tuned NMT; (3) ***k*NN-MT** stands for (Khandelwal et al., 2021) based on the Base MT; (4) ***k*NN-FT-MT** stands for the kNN algorithm applied to Fine-tuned MT, where the datastore is constructed based on fine-tuned model weights; (5) **Base *trainable-k*NN-MT** means fine-tuning with *trainable-k*NN-MT from the Base MT; (6) **FT *trainable-k*NN-MT** means continuous fine-tuning with *trainable-k*NN-MT from Fine-tuned MT. All related kNN-MT baseline systems listed in the report have been cited accordingly, with detailed explanations provided in Section 2.

4.3 Experimental Results

Results for Vanilla Inference Our findings from experiments using vanilla inference (without interpolation of the kNN predictions) are presented in Tables 1, 2 and 3. Across various domains in German-English, English-German, and English-Chinese, the *trainable-k*NN-MT method consistently outperforms the Fine-tuned MT in both Base and FT settings, highlighting its potential as an innovative fine-tuning approach. Notably, the

[3] Results in German-English and English-German are assessed in a case-sensitive, detokenized setting, while English-Chinese is evaluated in a tokenized setting. We adopt the same evaluation procedure for SacreBLEU as outlined by (Khandelwal et al., 2021).

[4] We compute the chrF score using a n-gram order of 6, a word n-gram order of 2, and a beta parameter set to 3.

	English-German					English-Chinese			
Model	**IT**	**Medical**	**Law**	**Koran**	**Avg.**	**Subtitle**	**News**	**Thesis**	**Avg.**
kNN-MT (Khandelwal et al., 2021)	36.44	49.74	55.73	25.45	41.84	29.72	32.83	33.28	31.94
kNN-FT-MT	40.68	53.28	58.91	32.61	46.37	34.79	39.41	56.49	43.56
Base *trainable-k*NN-MT (Ours)	41.04	53.14	58.89	32.44	46.38	34.62	**41.41**	**63.93**	**46.65**
FT *trainable-k*NN-MT (Ours)	**41.68**	**54.37**	**58.99**	**32.92**	**46.99**	**36.26**	39.83	57.04	44.38

Table 5: Performance of the *trainable-k*NN-MT for **kNN inference** on **English-German** and **English-Chinese** domain-specific test sets, measured in SacreBLEU. The *trainable-k*NN-MT significantly outperforms kNN-FT-MT by 0.62 BLEU and 3.09 BLEU for English-German and English-Chinese mutli-domain translations respectively.

FT *trainable-k*NN-MT model either significantly surpasses or achieves the comparable performance of the Base *trainable-k*NN-MT.

In German-English translations, the Base *trainable-k*NN-MT enhances performance by up to 0.72 BLEU and 0.58 chrF on average, while the FT *trainable-k*NN-MT boosts it by as much as 1.39 BLEU and 1.03 chrF. In English-German translations, the Base *trainable-k*NN-MT and FT *trainable-k*NN-MT surpass the Fine-tuned MT by the improvements of up to 0.93 BLEU and 0.76 chrF, and 1.90 BLEU and 1.34 chrF, respectively. For English-Chinese translations, the increases are up to 0.78 BLEU and 1.11 chrF for Base *trainable-k*NN-MT, and 1.31 BLEU and 1.21 chrF for FT *trainable-k*NN-MT. The Koran domain is the only case where *trainable-k*NN-MT cannot significantly outperform Fine-tuned MT. This could be due to the reason of data sparsity.

Results for kNN Inference When integrating the kNN predictions during inference, the *trainable-k*NN-MT consistently excels across multiple domains, as detailed in Tables 4 and 5. It's evident that either Base *trainable-k*NN-MT or FT *trainable-k*NN-MT surpass other kNN-related methods, highlighting the pivotal role of enhanced contextual representations in optimizing the kNN approach.

In details, for German-English, Base *trainable-k*NN-MT outperforms the original kNN-MT and kNN-FT-MT by as much as 2.85 BLEU and 0.35 BLEU, while FT *trainable-k*NN-MT yields improvements of 3.34 BLEU and 0.84 BLEU; for English-German, Base *trainable-k*NN-MT significantly outperforms the kNN-MT by 4.54 BLEU, but performs comparably with kNN-FT-MT, while the FT *trainable-k*NN-MT achieves improvements of 5.15 BLEU and 0.62 BLEU; for English-Chinese, Base *trainable-k*NN-MT surpasses the kNN-MT and kNN-FT-MT by 14.71 BLEU and 3.09 BLEU, while FT *trainable-k*NN-MT achieves improvements of 12.44 BLEU and 0.82 BLEU.

The hyper-parameters for the kNN search, utilized in Base *trainable-k*NN-MT models, are optimized based on the base NMT model as kNN-MT does, as detailed in Tables 9, 10, and 11 in Appendix A. These settings may not be optimal for models fine-tuned using the *trainable-k*NN-MT methods.

Interestingly, we observe that for German-English and English-German translations, there is a notable trend that the performance gap between fine-tuned models with and without kNN search integration during inference has narrowed when the model is tuned with *trainable-k*NN-MT. For example, for German-English translations, the gap between conventional Fine-tuned MT and its corresponding kNN-FT-MT is $48.34 - 47.09 = 1.25$ BLEU, while the gap between vanilla and kNN inference based on FT *trainable-k*NN-MT is $49.18 - 48.48 = 0.70$ BLEU. Given the slower inference speeds associated with kNN inference, this reduction in the performance gap is encouraging. It suggests that models tuned with *trainable-k*NN-MT can achieve impressive performance and inference efficiency without the need for kNN search integration.

4.4 Cross Domain Evaluation

Fine-tuning inherently carries the risk of overfitting. It is crucial to assess the *trainable-k*NN-MT's capacity to generalize to new domains after fine-tuning. One effective approach is through cross-domain evaluation. We've performed experiments in three language pairs using vanilla inference, with results presented in Tables 12, 13, and 14 in Appendix A. These results reveal that our proposed *trainable-k*NN-MT either outperform or perform on par with conventional fine-tuning. Informed by the ground truth probability within the kNN predictions, the dynamic gradient adjustment facilitates the NMT model to prioritize learning in areas of deficiency and deemphasize areas of expertise, potentially helping mitigate overfitting. Notably, the Base *trainable-*kNN-MT surpasses the FT *trainable-*kNN-MT in cross-domain evaluation except for the Koran domain, likely due to its adaptation from a foundational state pre-trained on out-of-domain data, whereas the latter continues tuning a model that has already undergone adjustments via conventional fine-tuning.

4.5 Training Efficiency

During training, we follow (Khandelwal et al., 2021) to use FAISS (Johnson et al., 2021) index for the kNN search, with which the keys are stored in clusters for searching speed-up, and quantized to 64-bytes for space efficiency. The indexes are constructed offline via forward passes over every example in the in-domain training set. While this approach conserves space and boosts search speed, the *trainable-*kNN-MT still requires more training time, as illustrated in Table 15 in Appendix A. Fortunately, this extended time only impacts training; during inference, we can opt for the vanilla inference without the kNN search when time efficiency is paramount.

4.6 Mixed-domain Adaptation

Finally, we assess the effectiveness of *trainable-*kNN-MT on a mixed-domain setting to determine if there is a consistent enhancement. Specifically, we combine the IT, Medical, Law, and Koran in-domain data from the German-English datasets to generate mixed-domain train/validation/test sets and investigate the performance of FT *trainable-*kNN-MT. We initially fine-tune the base NMT model on the mixed-domain data until it reaches convergence with an early stop mechanism of 30 epochs, and then we use *trainable-*kNN-MT for continuous fine-tuning for 30 epochs. Results are displayed in Table 16 in Appendix A, and indicate that compared with mixed-domain fine-tuning, FT *trainable-*kNN-MT brings an improvement of 0.85 SacreBLEU.

5 Conclusion

We present the *trainable-*kNN-MT, combining the kNN method with fine-tuning to enhance domain adaptation for neural machine translation. It leverages the ground truth probability within the kNN prediction to dynamically adjust gradients during fine-tuning. Experimentally, the *trainable-*kNN-MT surpasses conventional fine-tuning in domain-specific translations, helps mitigate overfitting, and augments the kNN-MT algorithm with improved contextual representations.

6 Limitations

The design of the current method is subject to limitations. Firstly, the training time efficiency of the *trainable-*kNN-MT requires future optimization. Accelerating the search process might be achievable by exploring fewer clusters or querying smaller datastores. Secondly, for low-resourced translations, achieving similar domain adaptation improvements with *trainable-*kNN-MT is challenging. Data sparsity becomes more pronounced, raising questions about the efficacy of kNN predictions in aiding NMT model learning. Further investigations into low-resourced translations are warranted.

Nonetheless, we would like to emphasize the significance of our work in the era of LLMs. Compared with conventional fine-tuning, the statistical insights from the kNN retrievals, as highlighted by our proposed approaches, can yield further improvements in fine-tuning and help alleviate overfitting. Significantly, these techniques are versatile and can be integrated with any generative models - including LLMs. To provide context, even before the advent of kNN-MT, the kNN method had showcased its value in language modeling, a point we touched upon in our introduction, referencing (Khandelwal et al., 2020). Within the realm of neural machine translation models adopting an encoder-decoder framework, we've identified the strengths of our techniques. This paves the way for deeper exploration into their utility and efficacy when applied to LLMs with a decoder-only design, probing further for potential performance enhancements through the supervised fine-tuning (SFT) of LLMs.

References

Aharoni, Roee and Yoav Goldberg. 2020. Unsupervised domain clusters in pretrained language models. *arXiv preprint arXiv:2004.02105*.

Barone, Antonio Valerio Miceli, Barry Haddow, Ulrich Germann, and Rico Sennrich. 2017. Regularization techniques for fine-tuning in neural machine translation. *arXiv preprint arXiv:1707.09920*.

Cao, Zhiwei, Baosong Yang, Huan Lin, Suhang Wu, Xiangpeng Wei, Dayiheng Liu, Jun Xie, Min Zhang, and Jinsong Su. 2023. Bridging the domain gaps in context representations for k-nearest neighbor neural machine translation. *arXiv preprint arXiv:2305.16599*.

Chu, Chenhui and Rui Wang. 2018. A survey of domain adaptation for neural machine translation. *arXiv preprint arXiv:1806.00258*.

Chu, Chenhui, Raj Dabre, and Sadao Kurohashi. 2017. An empirical comparison of domain adaptation methods for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 385–391.

Jiang, Hui, Ziyao Lu, Fandong Meng, Chulun Zhou, Jie Zhou, Degen Huang, and Jinsong Su. 2022. Towards robust k-nearest-neighbor machine translation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5468–5477, Abu Dhabi, United Arab Emirates, December. Association for Computational Linguistics.

Johnson, Jeff, Matthijs Douze, and Hervé Jégou. 2021. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547.

Khandelwal, Urvashi, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Generalization through memorization: Nearest neighbor language models. In *International Conference on Learning Representations*.

Khandelwal, Urvashi, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2021. Nearest neighbor machine translation. In *International Conference on Learning Representations*.

Kingma, Diederik P. and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In Bengio, Yoshua and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Kirkpatrick, James, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.

Ng, Nathan, 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, August. Association for Computational Linguistics.

Ott, Myle, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Popović, Maja. 2015. chrf: character n-gram f-score for automatic mt evaluation. In *Proceedings of the tenth workshop on statistical machine translation*, pages 392–395.

Post, Matt. 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, October. Association for Computational Linguistics.

Saunders, Danielle. 2022. Domain adaptation and multi-domain adaptation for neural machine translation: A survey. *Journal of Artificial Intelligence Research*, 75:351–424.

Sennrich, Rico, 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, August. Association for Computational Linguistics.

Tian, Liang, Derek F Wong, Lidia S Chao, Paulo Quaresma, Francisco Oliveira, and Lu Yi. 2014. Um-corpus: A large english-chinese parallel corpus for statistical machine translation. In *LREC*, pages 1837–1842.

Vaswani, Ashish, 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.

Wang, Dexin, Kai Fan, Boxing Chen, and Deyi Xiong. 2022. Efficient cluster-based k-nearest-neighbor machine translation. *arXiv preprint arXiv:2204.06175*.

Yang, Zhixian, Renliang Sun, and Xiaojun Wan. 2022. Nearest neighbor knowledge distillation for neural machine translation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5546–5556, Seattle, United States, July. Association for Computational Linguistics.

Zheng, Xin, Zhirui Zhang, Junliang Guo, Shujian Huang, Boxing Chen, Weihua Luo, and Jiajun Chen. 2021. Adaptive nearest neighbor machine translation. *arXiv preprint arXiv:2105.13022*.

A Appendix

This appendix furnishes additional analysis and experimental results that complement the findings discussed in the main body of the paper.

Source	*Sie können Writer-Textrahmen so miteinander verketten, dass ihr Inhalt automatisch von einem Rahmen in den nächsten fließt.*								
Reference	*You can **link** Writer text frames so that their contents automatically flow from one frame to another.*								
MT Prefix	*You can*								
kNN Distribution based on Base MT	Sub. Prob.	*join* *0.379*	***link*** *0.242*	*chain* *0.103*	***link*** *0.093*	*use* *0.068*	***link*** *0.041*	*connect* *0.037*	*comb@@* *0.032*
kNN Distribution based on Fine-tuned MT	Sub. Prob.	***link*** *0.698*	*chain* *0.243*	*nest* *0.029*	*nest* *0.012*	***link*** *0.008*	***link*** *0.003*	***link*** *0.002*	***link*** *0.001*
Source	*Die Quell- und Zielansicht ist der Hauptarbeitsbereich von Kompare;*								
Reference	*The source and destination **view** is the main workspace of Kompare;*								
MT Prefix	*The source and destination*								
kNN Distribution based on Base MT	Sub. Prob.	*p@@* *0.679*	*ann@@* *0.110*	*brow@@* *0.049*	***view*** *0.039*	***view*** *0.035*	***view*** *0.034*	***view*** *0.025*	***view*** *0.025*
kNN Distribution based on Fine-tuned MT	Sub. Prob.	***view*** *0.361*	*View* *0.193*	***view*** *0.159*	*View* *0.073*	*View* *0.073*	*View* *0.048*	***view*** *0.047*	***view*** *0.043*

Table 6: In German-English IT translations, we provide examples of the kNN prediction distributions for the next word prediction. "Sub." denotes retrieved subwords and "Prob." represents their probabilities. "MT Prefix" refers to the generated prefix tokens. For comparison, we applied the kNN search on both base and fine-tuned NMT models, resulting in different query representations and datastore key states. The ground-truth target is highlighted in **bold**. In the top example, the aggregated probabilities for the ground-truth target from the kNN distributions are $0.242 + 0.093 + 0.041 = 0.376$ for Base MT and $0.698 + 0.008 + 0.003 + 0.002 + 0.001 = 0.712$ for Fine-tuned MT. Similarly, in the bottom example, the probabilities are 0.158 for Base MT and 0.610 for Fine-tuned MT. It is evident that with improved hidden states, the kNN prediction distributions align more closely with the ground truth targets.

Domain	IT	Medical	Law	Koran
Train	177,792	206,804	447,696	14,979
Validation	2,000	2,000	2,000	2,000
Test	2,000	2,000	2,000	2,000
De-En Datastore size	3.10M	5.70M	18.38M	0.45M
En-De Datastore size	3.33M	6.13M	18.77M	0.48M

Table 7: The datastore size and the number of the parallel sentence pairs of the training/validation/test sets in each domain for German-English (De-En) and English-German (En-De) domain-specific translations.

Domain	Subtitle	News	Thesis
Train	298,000	448,000	298,000
Validation	2,000	2,000	2,000
Test	597	1,500	625
En-Zh Datastore size	3.43M	14.32M	10.58M

Table 8: The datastore size and the number of the parallel sentence pairs of the training/validation/test sets in each domain for English-Chinese (En-Zh) domain-specific translations.

Model		IT	Medical	Law	Koran
kNN-MT & Base *trainable-*kNN-MT	k	8	16	16	8
	λ	0.6	0.8	0.8	0.6
	T	5	5	5	100
kNN-FT-MT & FT *trainable-*kNN-MT	k	8	16	8	8
	λ	0.4	0.4	0.6	0.4
	T	10	10	10	100

Table 9: The hyper-parameters used in the models for German-English translations.

Model		IT	Medical	Law	Koran
kNN-MT & Base *trainable-*kNN-MT	k	8	8	16	16
	λ	0.6	0.8	0.8	0.8
	T	10	10	5	10
kNN-FT-MT & FT *trainable-*kNN-MT	k	4	4	8	16
	λ	0.4	0.4	0.4	0.2
	T	10	100	5	5

Table 10: The hyper-parameters used in the models for English-German translations.

Model		Subtitle	News	Thesis
kNN-MT & Base *trainable-*kNN-MT	k	16	16	16
	λ	0.8	0.4	0.4
	T	10	10	10
kNN-FT-MT & FT *trainable-*kNN-MT	k	4	4	16
	λ	0.2	0.4	0.2
	T	5	100	10

Table 11: The hyper-parameters used in the models for English-Chinese translations.

		Evaluation				
		IT	Medical	Law	Koran	Avg.
Fine-tuned MT	IT	—	24.67	17.92	7.23	16.61
	Medical	19.78	—	25.46	3.49	16.24
	Law	18.23	25.84	—	2.95	15.67
	Koran	33.39	35.71	40.58	—	36.56
Base *trainable-*kNN-MT (Ours)	IT	—	27.87	24.39	9.42	20.56
	Medical	21.86	—	27.82	4.22	17.97
	Law	19.99	26.61	—	3.67	16.76
	Koran	33.21	36.04	40.33	—	36.53
FT *trainable-*kNN-MT (Ours)	IT	—	25.66	18.05	7.53	17.08
	Medical	20.40	—	26.19	3.82	16.80
	Law	19.51	26.56	—	3.05	16.37
	Koran	33.50	35.75	40.63	—	36.63

Table 12: Cross-domain evaluation of **German-English** translations using SacreBLEU. Models, fine-tuned on in-domain data (second column), are assessed on out-of-domain test sets (second row). Average SacreBLEU scores across these test sets suggest that both the Base *trainable-*kNN-MT and the FT *trainable-*kNN-MT generally surpass conventional fine-tuning in cross-domain evaluations, with the exception of the Koran domain where they exhibit comparable performance. Moreover, the Base *trainable-*kNN-MT outperforms the FT *trainable-*kNN-MT on average.

		Evaluation				
		IT	Medical	Law	Koran	Avg.
Fine-tuned MT	IT	—	21.96	16.99	5.25	14.73
	Medical	13.28	—	18.12	2.09	11.16
	Law	12.48	23.71	—	2.25	12.81
	Koran	12.25	13.64	19.73	—	15.21
Base *trainable-k*NN-MT (Ours)	IT	—	27.07	25.45	7.76	20.09
	Medical	17.27	—	20.42	3.00	13.56
	Law	14.02	25.19	—	2.66	13.96
	Koran	6.63	5.81	8.69	—	7.04
FT *trainable-k*NN-MT (Ours)	IT	—	23.11	17.60	5.75	15.49
	Medical	14.56	—	18.52	2.34	11.81
	Law	12.66	24.39	—	2.36	13.14
	Koran	17.60	15.80	20.80	—	18.07

Table 13: Cross-domain evaluation of **English-German** translations using SacreBLEU. Models, fine-tuned on in-domain data (second column), are assessed on out-of-domain test sets (second row). Average SacreBLEU scores across these test sets suggest that both the Base *trainable-k*NN-MT and the FT *trainable-k*NN-MT generally surpass conventional fine-tuning in cross-domain evaluations, with the exception of the Koran domain.

		Evaluation			
		Subtitles	News	Thesis	Avg.
Fine-tuned MT	Subtitles	—	4.16	1.11	2.64
	News	21.71	—	22.84	22.28
	Thesis	12.52	14.16	—	13.34
Base *trainable-k*NN-MT (Ours)	Subtitles	—	3.42	17.84	10.63
	News	24.84	—	27.18	26.01
	Thesis	14.77	17.54	—	16.16
FT *trainable-k*NN-MT (Ours)	Subtitles	—	4.37	1.29	2.83
	News	21.70	—	22.96	22.33
	Thesis	12.13	14.00	—	13.07

Table 14: Cross-domain evaluation of **English-Chinese** translations using SacreBLEU. Models, fine-tuned on in-domain data (second column), are assessed on out-of-domain test sets (second row). Average SacreBLEU scores across these test sets suggest that the Base *trainable-k*NN-MT typically exceeds conventional fine-tuning, while the FT *trainable-k*NN-MT matches conventional performance.

	IT	**Medical**	**Law**	**Koran**
Datastore Size	3.10M	5.70M	18.38M	0.45M
Cost of Conventional Fine-tuning	699.5s	1320.1s	4560.4s	173.2s
Cost of Datastore Construction of *trainable-k*NN-MT	290.0s	569.0s	1762.0s	105.0s
Cost of *trainable-k*NN-MT Training	1404.9s	3854.8s	11832.8s	377.6s
Total Cost of *trainable-k*NN-MT Training	1694.9s	4423.8s	13594.8s	482.6s

Table 15: Comparisons of training cost of one epoch between the *trainable-k*NN-MT and conventional fine-tuning for German-English domain-specific translations. The total parameters of the NMT model is 269,746,176 and the size of the hidden state is 1024. The training time for *trainable-k*NN-MT is 2 to 3 times longer than that of the conventional fine-tuning. In our implementation, we have adopted FAISS index (Johnson et al., 2021) for the *k*NN retrieval, which inherently incorporates the product quantization (PQ) as a foundational technique for vector compression. Within FAISS, PQ serves both to compress vectors and to expedite distance computations between a compressed vector and a full-sized vector, which enables our trainable-kNN-MT to efficiently handle large-scale similarity searches with millions of vectors.

Model	**IT**	**Medical**	**Law**	**Koran**	**Avg.**
Mixed Fine-tuned MT	46.82	57.43	61.04	21.36	46.66
FT *trainable-k*NN-MT (Ours)	47.79	58.05	62.96	21.24	47.51

Table 16: Performance of FT *trainable-k*NN-MT on German-English mixed-domain adaption with vanilla inference. FT *trainable-k*NN-MT exhibits an enhancement of 0.85 SacreBLEU points over the baseline established by the mixed-domain fine-tuning method.