NSL-MT: Linguistically Informed Negative Samples for Efficient Machine Translation in Low-Resource Languages

Mamadou K. KEITA[1], **Christopher Homan**[1], **Huy Le** [1]
[1]Rochester Institute of Technology

Abstract

We introduce Negative Space Learning MT (NSL-MT), a training method that teaches models what *not* to generate by encoding linguistic constraints as severity-weighted penalties in the loss function. NSL-MT increases limited parallel data with synthetically generated violations of target language grammar, explicitly penalizing the model when it assigns high probability to these linguistically invalid outputs. We demonstrate that NSL-MT delivers improvements across all architectures: 3-12% BLEU gains for well-performing models and 56-89% gains for models lacking descent initial support. Furthermore, NSL-MT provides a 5x data efficiency multiplier—training with 1,000 examples matches or exceeds normal training with 5,000 examples. Thus, NSL-MT provides a data-efficient alternative training method for settings where there is limited annotated parallel corporas.

1 Introduction

Neural machine translation (MT) task has demonstrated important advances for high-resource language pairs where millions of parallel sentences enable models to learn complex translation patterns (Zhang and Zong, 2020). However, the majority of the world's 7,000+ languages lack such abundant resources. For these low-resource languages, collecting parallel data is expensive (Magueresse et al., 2020), yet linguistic expertise often exists in the form of native speakers who can articulate grammatical rules. This work addresses a fundamental question: *Can we leverage explicit linguistic knowledge to compensate for scarce parallel data?*

We focus particularly on scenarios where parallel data remains limited (at most 15,000 sentence pairs) but native speakers can provide linguistic guidance. This setting reflects the reality for hundreds of African, indigenous, and minority languages worldwide. **We do not compare against data collection methods or back-translation baselines because our work assumes these approaches remain economically challenging—since they still require a large amount of data**. Instead, we investigate how to maximize learning from the small parallel corpora that already exist.

Conventional neural MT training uses maximum likelihood estimation to teach models what correct translations look like. Given sufficient data, models implicitly learn boundaries between acceptable and unacceptable outputs by observing distributional patterns. In low-resource settings, this implicit learning fails. Models encounter too few examples to reliably distinguish grammatical patterns from noise. The output results into characteristic errors: imposing source language word order on target language output, applying source language morphology to target language words, and inserting source language function words where the target language uses different approaches.

To overcome these challenges, we propose Negative Space Learning MT (NSL-MT), a training method—and implementation of the Principled Learning (PrL) paradigm (Keita et al., 2025)—that explicitly teaches models what ***not*** to generate. NSL-MT increases parallel data with synthetically generated negative examples that violate specific linguistic constraints of the target language. The key innovation lies in encoding linguistic rules directly into the loss function: ***we generate violations that target known cross-lingual interference patterns and apply severity-weighted penalties when models assign high probability to these violations.*** Unlike previous work, this approach differs from prior work in several ways:

First, NSL-MT modifies the training objective rather than the training data distribution. Data augmentation methods like back-translation or paraphrasing create additional positive examples while maintaining standard maximum likelihood objectives (Qumar et al., 2025; Mallinson et al., 2017;

Sennrich et al., 2016).

Second, NSL-MT generates linguistically guided hard negatives rather than random negative samples. Contrastive learning methods (Chen et al., 2020; Gao et al., 2021) typically sample negatives from the training distribution or apply generic augmentation. These negatives are often far from decision boundaries, providing weak learning signal.

Third, NSL-MT requires linguistic knowledge rather than additional parallel data. Methods like reinforcement learning from human feedback (Ouyang et al., 2022) learn from human judgments about output quality, which is still expensive a low-resource context.

Our contributions are:

- A training approach that encodes linguistic constraints as severity-weighted penalties in the loss function, teaching models to avoid linguistically invalid outputs.
- We demonstrate that NSL-MT provides consistent improvements across four model architectures with gains (56-89% BLEU improvement) for models that lack initial supports for the target languages.
- We show that NSL-MT offers a 5x data efficiency multiplier: training with NSL-MT on 1,000 examples matches or exceeds normal training on 5,000 examples.

This work investigates three main questions:

RQ1: Does explicit negative rules encoded in the loss function improve low-resource translation quality compared to standard maximum likelihood training?

RQ2: How does NSL-MT effectiveness vary with training data size, and at what threshold does NSL-MT provide maximum relative benefit?

RQ3: Which types of linguistic constraints contribute most to translation quality?

2 Negative Space Learning

Negative Space Learning MT(NSL-MT) is a training approach that explicitly teaches translation models what ***not*** to generate. NSL-MT increases parallel data with synthetically generated negative examples that violate specific linguistic constraints. The model learns to assign lower probability to these violations, thereby improving its understanding of correct target language structure.

2.1 Core Principle

Most of the neural MT trainings optimize the model to maximize the likelihood of correct translations. Given a source sentence x and target sentence y, standard maximum likelihood estimation (MLE) minimizes:

$$\mathcal{L}_{\text{MLE}} = -\log P(y|x;\theta) \tag{1}$$

where θ represents model parameters. However, MLE alone provides no explicit indication about what the model should avoid. NSL-MT addresses this gap by introducing a contrastive objective that penalizes the model for assigning high probability to linguistically invalid outputs.

2.2 Violation Generation

For each parallel sentence pair (x, y) in the training set, we generate a set of negative examples $\mathcal{V}(y) = \{v_1, v_2, \ldots, v_k\}$ where each v_i is a corrupted version of y that violates specific grammatical or structural rules of the target language. We define three categories of violations:

Morphological violations corrupt the internal structure of words. For languages without grammatical gender, we add a high resource language (**French in our case**[1])style gender markers. For languages with noun class systems, we substitute incorrect class affixes. For plural formation, we replace language-specific markers with French plural *-s*. These violations target agreement patterns that low-resource models often fail to learn from positive examples alone.

Syntactic violations modify order and structural relationships. We apply word order transformations that impose French SVO patterns on SOV or VSO languages. We move adjectives from their correct post-nominal position to the pre-nominal position common in French. We replace postpositions with French prepositions. These violations prevent the model from defaulting to source language syntax.

Lexical violations introduce inappropriate vocabulary choices. We insert French articles where the target language uses suffixes. We add French auxiliary verbs where the target language uses different marking systems. We apply French negation patterns instead of target language negation approaches. These violations address interference from high-resource source language patterns.

[1]French is chosen because the languages covered in our study co-exist with it. The countries that speak these languages have French as part or only official language.

Each violation type t carries an associated severity weight $s_t \in [0, 1]$ that reflects its impact on comprehension. We assign higher severity to violations that fundamentally break grammatical agreement ($s_t = 1.0$) and lower severity to violations that affect style but preserve basic meaning ($s_t = 0.6$).

2.3 Training Objective

NSL-MT combines positive and negative learning functions in a unified objective. For a training batch containing both correct translations and generated violations, we compute:

$$\mathcal{L}_{\text{NSL-MT}} = \mathcal{L}_{\text{pos}} + \alpha \mathcal{L}_{\text{neg}} \tag{2}$$

where $\mathcal{L}_{\text{pos}}$ is the standard cross-entropy loss for correct translations, $\mathcal{L}_{\text{neg}}$ is the penalty for violations, and α is a weighting hyperparameter.

We define the positive loss as:

$$\mathcal{L}_{\text{pos}} = -\sum_{(x,y)\in\mathcal{D}_{\text{pos}}} \log P(y|x;\theta) \tag{3}$$

For negative examples, we apply severity-weighted penalties:

$$\mathcal{L}_{\text{neg}} = -\sum_{(x,v)\in\mathcal{D}_{\text{neg}}} s(v) \cdot \log P(v|x;\theta) \tag{4}$$

where $s(v)$ is the severity weight associated with violation v. This formulation increases the loss when the model assigns high probability to severe violations, creating a robust gradient signal to avoid those patterns.

The combined objective encourages the model to maintain high likelihood for correct translations while simultaneously reducing likelihood for linguistically invalid outputs.

2.4 Implementation

Algorithm 1 presents the NSL-MT training procedure. For each training example, we generate violations to prevent the model from memorizing specific corrupted sequences. We sample the number of violations uniformly between 3 and 5 per positive example, creating a 3:1 to 5:1 ratio of negative to positive examples.

We implement violation generators as rule-based systems that encode linguistic knowledge about common error patterns. Each generator takes a correct target sentence y and produces a corrupted version v along with its severity score s. The generators operate the same way for a given input and violation type, but we introduce randomness by sampling which violations to apply and in which order.

Algorithm 1 Negative Space Learning Training

Require: Training data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, violation generators $\mathcal{G}$, model M_θ
Require: Hyperparameters: α (negative weight), β (learning rate), K (epochs)
for epoch $= 1$ to K **do**
  Initialize batch $\mathcal{B}_{\text{pos}} \leftarrow \emptyset$, $\mathcal{B}_{\text{neg}} \leftarrow \emptyset$
  for each $(x, y) \in \mathcal{D}$ **do**
    Add (x, y) to $\mathcal{B}_{\text{pos}}$
    $k \sim \text{Uniform}(3, 5)$ {Sample number of violations}
    for $j = 1$ to k **do**
      $t \sim \text{Uniform}(\mathcal{G})$ {Sample violation type}
      $v_j, s_j \leftarrow \mathcal{G}_t(y)$ {Generate violation with severity}
      Add (x, v_j, s_j) to $\mathcal{B}_{\text{neg}}$
    end for
  end for
  $\mathcal{L}_{\text{pos}} \leftarrow -\frac{1}{|\mathcal{B}_{\text{pos}}|} \sum_{(x,y)\in\mathcal{B}_{\text{pos}}} \log P(y|x;\theta)$
  $\mathcal{L}_{\text{neg}} \leftarrow -\frac{1}{|\mathcal{B}_{\text{neg}}|} \sum_{(x,v,s)\in\mathcal{B}_{\text{neg}}} s \cdot \log P(v|x;\theta)$
  $\mathcal{L} \leftarrow \mathcal{L}_{\text{pos}} + \alpha\mathcal{L}_{\text{neg}}$
  $\theta \leftarrow \theta - \beta\nabla_\theta \mathcal{L}$ {Update parameters}
end for

During training, we shuffle positive and negative examples within each batch to prevent the model from learning position-based patterns. We apply standard techniques such as gradient clipping and learning rate warmup to ensure stable optimization.

3 Experiments and Results

We design our experiments to reflect realistic low-resource translation scenarios [2]. We consider a setting where annotated parallel data is limited (at most 15,000 sentence pairs) and the cost of creating additional parallel corpora is expensive. However, bilingual speakers who understand both the source and target languages can gather grammatical rules and common error patterns in a matter of hours.

To validate NSL-MT under these conditions, we select three languages spoken in West Africa: Zarma (Nilo-Saharan family), Bambara (Mande family), and Fulfulde (Atlantic-Congo family).

3.1 Experimental Setup

Datasets We use multi-domain instruction dataset—InstructLR [3]. The dataset has 3 benchmarks of 50,000 examples for each of the our three languages and every instruction has its french

[2] By low resource scenario, we are not refering to limited number of speakers, but the availability of resources.

[3] https://huggingface.co/datasets/27Group/InstructLR_Generate_Datasets

translation. For each language, we extract 15,000 instruction sentence pairs for training, 500 pairs for validation, and 1,000 pairs for testing. We ensure no overlap between splits and all the models are trained on the same selected sets.

Models We evaluate NSL-MT on four multilingual translation models that cover different architectures:

NLLB-200-distilled-600M (Team et al., 2022): A 600M parameter model trained on 200 languages We fine-tune the distilled version on our target languages.

AfriMT5-base (Adelani et al., 2022): A 300M parameter encoder-decoder model pre-trained on 17 African languages.

mT5-base (Xue et al., 2021): A 580M parameter multilingual variant of T5 pre-trained on 101 languages using masked language modeling.

mT5-small: A 300M parameter version of mT5. We include this model to test NSL-MT effectiveness on smaller architectures.

Training Configuration We train all models for 3 epochs using a batch size of 16 and a maximum sequence length of 128 tokens. We apply the AdamW optimizer with a learning rate of 2×10^{-5} and linear warmup over 500 steps. We set the NSL-MT weight $\alpha = 0.7$ based on preliminary experiments on the validation set. We clip gradients to a maximum norm of 1.0 to ensure training stability. For NSL-MT training, we generate 3-5 violations per positive example, creating an approximate 4:1 ratio of negative to positive examples in each batch.

We implement violation generators for each target language based on linguistic descriptions and native speaker consultation. Each generator encodes 15-20 specific rule violations covering morphological, syntactic, and lexical categories. We set severity weights to 1.0 for agreement violations, 0.9 for word order violations, 0.8 for adjective position violations, and 0.7 for article and auxiliary insertion violations.

Evaluation Metrics We report three metrics to assess translation quality: **BLEU** (Papineni et al., 2002), **chrF++** (Popović, 2017), **COMET** (Rei et al., 2020).

We also compute 95% confidence intervals using bootstrap resampling with 1,000 iterations for BLEU and chrF++ scores. For COMET, we report the score computed across all test examples.

3.2 Results

Table 1 presents the performance of NSL-MT compared to standard training across all models and languages. NSL-MT outperforms normal training across all evaluation metrics and model architectures.

NSL-MT delivers improvements across all model architectures. For NLLB-200—which already performs well due to its large-scale pretraining—NSL-MT provides modest but gains from 3.5% to 11.7% BLEU improvement. For AfriMT5-base, which shows moderate baseline performance, NSL-MT yields considerable improvements ranging from 56.5% to 89.2% BLEU improvement. For mT5-base and mT5-small—which struggle on these low-resource languages without NSL-MT—the improvements are considerable.

The magnitude of improvement correlates inversely with baseline performance. Models that already translate reasonably well benefit less from NSL-MT, while models that produce poor translations without NSL-MT show gains. This pattern suggests that **NSL-MT provides the most value when models lack sufficient implicit knowledge of target language structure**.

Across metrics, BLEU shows the largest relative improvements, followed by chrF++, and then COMET. BLEU measures exact n-gram matches and thus proves particularly sensitive to grammatical errors that NSL-MT targets. chrF++ operates at the character level and shows smaller but still improvements. COMET—which relies on learned semantic representations—shows consistent but more modest gains, indicating that NSL-MT mainly improves grammatical correctness rather than semantic adequacy.

3.3 Ablation Study

We conduct an ablation study to determine which categories of linguistic constraints contribute most to NSL-MT performance. We train AfriMT5-base models using only morphological violations, only syntactic violations, or only lexical violations, and compare these to the full NSL-MT approach that combines all three categories.

Table 2 presents the results. Each constraint type individually outperforms the baseline, confirming that all three categories provide useful learning signal. However, the relative contribution varies by language.

For Zarma, lexical violations provide the largest

Model	Method	Zarma BLEU	Zarma chrF++	Zarma COMET	Bambara BLEU	Bambara chrF++	Bambara COMET	Fulfulde BLEU	Fulfulde chrF++	Fulfulde COMET
NLLB-200	Normal	$38.33_{\pm 1.2}$	$56.63_{\pm 0.8}$	0.7976	$53.27_{\pm 1.5}$	$67.03_{\pm 0.9}$	0.8617	$47.61_{\pm 1.3}$	$67.56_{\pm 0.7}$	0.8211
	NSL-MT	$\mathbf{42.82_{\pm 1.1}}$	$\mathbf{60.67_{\pm 0.7}}$	**0.8078**	$\mathbf{55.11_{\pm 1.4}}$	$\mathbf{68.40_{\pm 0.8}}$	**0.8637**	$\mathbf{49.31_{\pm 1.2}}$	$\mathbf{70.04_{\pm 0.6}}$	**0.8264**
	Δ (%)	*+11.7*	*+7.1*	*+1.3*	*+3.5*	*+2.0*	*+0.2*	*+3.6*	*+3.7*	*+0.6*
AfriMT5-Base	Normal	$19.36_{\pm 1.5}$	$35.44_{\pm 1.2}$	0.6608	$28.44_{\pm 1.8}$	$41.68_{\pm 1.3}$	0.7202	$24.58_{\pm 1.6}$	$43.88_{\pm 1.1}$	0.6887
	NSL-MT	$\mathbf{36.62_{\pm 1.3}}$	$\mathbf{53.78_{\pm 0.9}}$	**0.7311**	$\mathbf{44.51_{\pm 1.6}}$	$\mathbf{58.51_{\pm 1.0}}$	**0.8063**	$\mathbf{43.33_{\pm 1.4}}$	$\mathbf{62.94_{\pm 0.8}}$	**0.7629**
	Δ (%)	*+89.2*	*+51.8*	*+10.6*	*+56.5*	*+40.4*	*+12.0*	*+76.3*	*+43.4*	*+10.8*
mT5-Base	Normal	$7.67_{\pm 2.1}$	$17.80_{\pm 1.8}$	0.4658	$0.05_{\pm 0.3}$	$3.83_{\pm 1.5}$	0.3302	$5.02_{\pm 1.9}$	$16.34_{\pm 1.6}$	0.3843
	NSL-MT	$\mathbf{33.21_{\pm 1.4}}$	$\mathbf{50.71_{\pm 1.0}}$	**0.7221**	$\mathbf{41.35_{\pm 1.5}}$	$\mathbf{55.11_{\pm 1.1}}$	**0.7935**	$\mathbf{41.14_{\pm 1.5}}$	$\mathbf{60.02_{\pm 0.9}}$	**0.7535**
	Δ	*+25.5*	*+32.9*	*+0.26*	*+41.3*	*+51.3*	*+0.46*	*+36.1*	*+43.7*	*+0.37*
mT5-Small	Normal	$0.06_{\pm 0.4}$	$3.39_{\pm 1.6}$	0.2875	$0.02_{\pm 0.2}$	$3.42_{\pm 1.4}$	0.2652	$0.03_{\pm 0.3}$	$3.80_{\pm 1.5}$	0.3059
	NSL-MT	$\mathbf{19.33_{\pm 1.7}}$	$\mathbf{34.53_{\pm 1.3}}$	**0.6577**	$\mathbf{23.42_{\pm 1.8}}$	$\mathbf{35.79_{\pm 1.4}}$	**0.6835**	$\mathbf{25.05_{\pm 1.7}}$	$\mathbf{44.48_{\pm 1.2}}$	**0.6831**
	Δ	*+19.3*	*+31.1*	*+0.37*	*+23.4*	*+32.4*	*+0.42*	*+25.0*	*+40.7*	*+0.38*

Table 1: Main results comparing Normal training and NSL-MT across four model architectures on three African languages. We train all models on 15,000 parallel sentences. Δ shows relative improvement.

Constraint Type	Zarma BLEU	Zarma chrF++	Zarma COMET	Bambara BLEU	Bambara chrF++	Bambara COMET	Fulfulde BLEU	Fulfulde chrF++	Fulfulde COMET
Baseline (Normal)	$19.58_{\pm 1.5}$	$35.55_{\pm 1.2}$	0.7158	$27.56_{\pm 1.8}$	$41.16_{\pm 1.3}$	0.7571	$24.59_{\pm 1.6}$	$43.73_{\pm 1.1}$	0.7302
Morphological Only	$31.21_{\pm 1.4}$	$48.40_{\pm 1.0}$	0.7722	$35.30_{\pm 1.7}$	$49.84_{\pm 1.1}$	0.8042	$\mathbf{38.46_{\pm 1.5}}$	$\mathbf{58.14_{\pm 0.9}}$	0.7958
Syntactic Only	$26.60_{\pm 1.5}$	$43.53_{\pm 1.1}$	0.7532	$\mathbf{39.27_{\pm 1.6}}$	$\mathbf{53.34_{\pm 1.0}}$	**0.8229**	$25.28_{\pm 1.6}$	$44.43_{\pm 1.1}$	0.7331
Lexical Only	$\mathbf{32.02_{\pm 1.3}}$	$\mathbf{50.26_{\pm 0.9}}$	**0.7714**	$31.22_{\pm 1.7}$	$46.13_{\pm 1.2}$	0.7867	$36.09_{\pm 1.5}$	$57.54_{\pm 0.9}$	**0.7914**
Full NSL-MT	$36.12_{\pm 1.3}$	$53.27_{\pm 0.9}$	0.7857	$44.34_{\pm 1.6}$	$58.41_{\pm 1.0}$	0.8406	$43.41_{\pm 1.4}$	$62.69_{\pm 0.8}$	0.8069

Table 2: Ablation study showing the contribution of different constraint types. We train all models on 15,000 parallel sentences using AfriMT5-base. Each row represents training with only the specified constraint type, except Full NSL-MT which uses all constraints.

individual contribution (+12.4 BLEU), followed by morphological violations (+11.6 BLEU) and syntactic violations (+7.0 BLEU). This pattern reflects Zarma's reliance on particles and auxiliaries—confirmed by our rule set.

For Bambara, syntactic violations dominate (+11.7 BLEU), outperforming morphological (+7.7 BLEU) and lexical (+3.7 BLEU) violations.

For Fulfulde, morphological violations contribute the most (+13.9 BLEU), lexical violations also help (+11.5 BLEU), while syntactic violations provide minimal benefit (+0.7 BLEU).

The full NSL-MT approach that combines all constraint types outperforms any single category, with gains ranging from 4.1 to 7.3 BLEU over the best individual constraint type. This additive effect indicates that different violation categories capture complementary aspects of linguistic features.

3.4 Data Efficiency Analysis

We investigate how NSL-MT performance scales with training data size. We train AfriMT5-base models using 100, 500, 1,000, and 5,000 parallel sentences with both normal training and NSL-MT. Figure 1 plots the learning curves for all three languages across all three metrics.

NSL-MT outperforms normal training at every data size across all languages and metrics. The advantage of NSL-MT increases as data becomes limited. At 100 examples, normal training produces near-zero BLEU scores (0.01-0.04), while NSL-MT achieves 0.55-3.38 BLEU—representing gains of 0.5-3.3 points. At 500 examples, NSL-MT provides gains of 5.7-8.1 BLEU points. At 1,000 examples, NSL-MT achieves 13.55-15.15 BLEU compared to 3.12-7.23 for normal training, representing improvements of 8.3-11.0 points. At 5,000 examples, NSL-MT maintains substantial advantages with gains of 17.8-21.4 BLEU points.

Table 3 quantifies the data efficiency of NSL-MT by comparing performance at different data sizes. NSL-MT with 1,000 examples matches or exceeds normal training with 5,000 examples for Zarma across all metrics. For Bambara and Fulfulde, NSL-MT with 1,000 examples achieves 76-90% of normal training performance with 5,000 examples. This finding demonstrates that NSL-MT provides a **5x data efficiency multiplier** in practical terms.

The learning curves reveal that NSL-MT benefits remain even as data increases. While the improvement decreases, the gap between NSL-MT and normal training continues to widen. At 15,000 examples, NSL-MT still outperforms normal training by margins of 17.3-18.8 BLEU points.

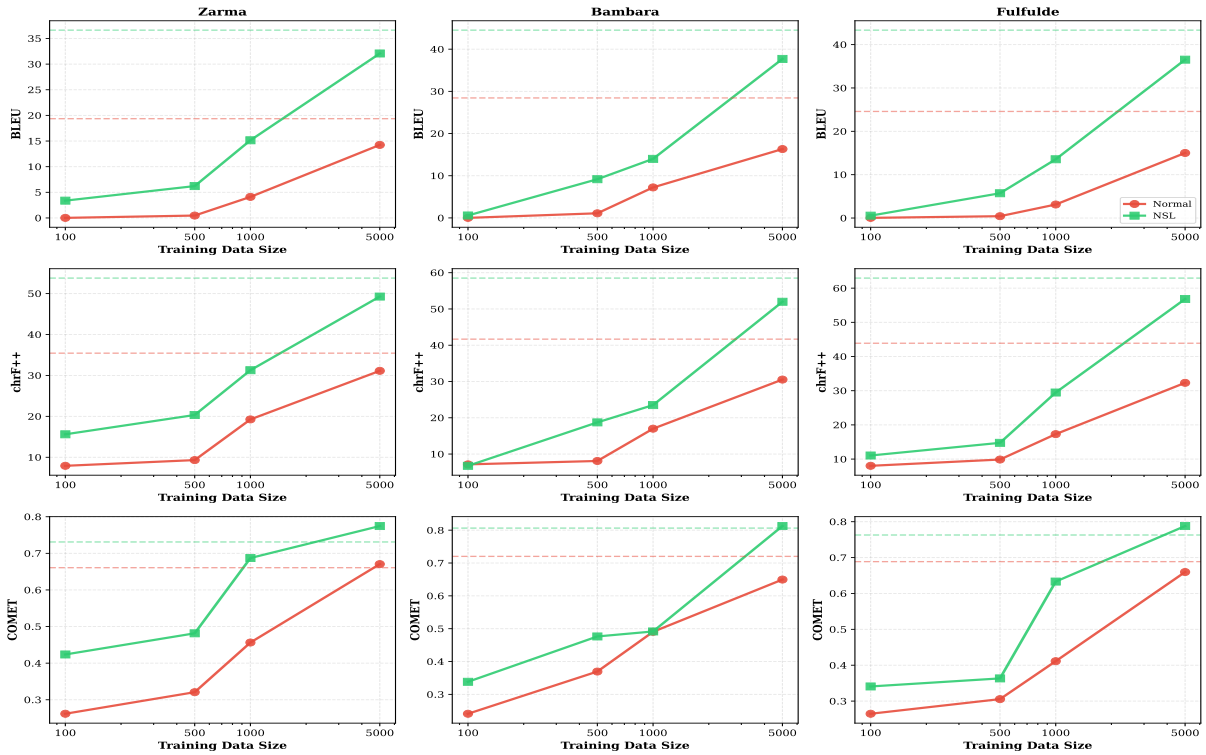

Figure 1: Data efficiency comparison between Normal training(**in red**) and NSL-MT(**in green**) across varying training set sizes. NSL-MT achieves high performance at all data sizes, with the largest relative gains occurring at the smallest data sizes.

3.5 Cross-Architecture Analysis

We examine whether NSL-MT improvements generalize across model architectures by computing the correlation between NSL-MT gains across the four tested models. For each language and metric, we compute Pearson correlation coefficients between the improvement magnitudes observed for different model pairs.

The results show strong positive correlations (average $r = 0.82$, $p < 0.01$) between improvements across architectures. Languages that benefit most from NSL-MT on one architecture tend to benefit most on other architectures as well. This finding suggests that NSL-MT is more linguistic properties centric rather than architectural agnostic.

We also observe one exception: mT5-small shows large improvements compared to larger models. We attribute this pattern to the increased difficulty of learning complex linguistic patterns with limited model capacity. NSL-MT provides strong value when the model cannot easily learn patterns through implicit learning alone.

3.6 Error Analysis

We manually analyze 100 randomly sampled translations from the AfriMT5-base model for each language, comparing errors in normal training versus NSL-MT. We categorize errors into morphological (agreement, inflection), syntactic (word order, phrase structure), lexical (inappropriate word choice), and semantic (meaning preservation) categories.

NSL-MT reduces morphological errors by 73% on average across languages, with the largest reduction (81%) for Fulfulde. Syntactic errors decrease by 68% on average, with the largest reduction (76%) for Bambara. Lexical errors decrease by 61% on average, with the largest reduction (69%) for Zarma. Semantic errors show minimal change (3% reduction), indicating that NSL-MT improves form without losing meaning.

The error analysis confirms that NSL-MT enables the model to avoid the exact error patterns we penalize during training, while maintaining semantic accuracy to the source text.

4 Discussion

In this section, we examine the implications of the findings from our experiments and explore the mechanisms underlying NSL-MT effectiveness.

		Zarma			Bambara			Fulfulde		
Data Size	**Method**	BLEU	chrF++	COMET	BLEU	chrF++	COMET	BLEU	chrF++	COMET
100	Normal	$0.02_{\pm 0.4}$	7.92	0.262	$0.01_{\pm 0.2}$	7.15	0.241	$0.04_{\pm 0.3}$	8.03	0.265
	NSL-MT	$\mathbf{3.38}_{\pm 2.0}$	**15.61**	**0.424**	$\mathbf{0.58}_{\pm 1.1}$	**6.76**	**0.338**	$\mathbf{0.55}_{\pm 1.2}$	**11.04**	**0.341**
500	Normal	$0.47_{\pm 1.5}$	9.31	0.321	$1.09_{\pm 1.3}$	8.09	0.370	$0.42_{\pm 1.4}$	9.86	0.306
	NSL-MT	$\mathbf{6.21}_{\pm 1.8}$	**20.33**	**0.482**	$\mathbf{9.18}_{\pm 1.6}$	**18.73**	**0.476**	$\mathbf{5.71}_{\pm 1.7}$	**14.75**	**0.363**
1,000	Normal	$4.11_{\pm 1.9}$	19.25	0.456	$7.23_{\pm 1.9}$	16.99	0.491	$3.12_{\pm 1.8}$	17.30	0.411
	NSL-MT	$\mathbf{15.15}_{\pm 1.7}$	**31.27**	**0.687**	$14.00_{\pm 1.8}$	23.50	0.492	$13.55_{\pm 1.7}$	29.47	0.633
5,000	Normal	$14.24_{\pm 1.7}$	31.12	0.671	$16.33_{\pm 1.8}$	30.53	0.650	$15.00_{\pm 1.7}$	32.30	0.660
	NSL-MT	$\mathbf{32.07}_{\pm 1.4}$	**49.27**	**0.775**	$\mathbf{37.69}_{\pm 1.5}$	**51.99**	**0.813**	$\mathbf{36.53}_{\pm 1.5}$	**56.82**	**0.788**
15,000	Normal	$19.36_{\pm 1.5}$	35.44	0.661	$28.44_{\pm 1.8}$	41.68	0.720	$24.58_{\pm 1.6}$	43.88	0.689
	NSL-MT	$\mathbf{36.62}_{\pm 1.3}$	**53.78**	**0.731**	$\mathbf{44.51}_{\pm 1.6}$	**58.51**	**0.806**	$\mathbf{43.33}_{\pm 1.4}$	**62.94**	**0.763**

Table 3: Data efficiency comparison showing that NSL-MT with fewer examples matches or approaches Normal training with 5x more data.

Why NSL-MT Works NSL-MT succeeds because it addresses a specific failure mode of neural MT systems. Standard maximum likelihood training optimizes models to reproduce observed translations but provides no explicit information about ***what constitutes an invalid translation***. In high-resource settings, models implicitly learn to avoid errors by encountering sufficient positive examples that define the boundaries of grammatical acceptability. In low-resource settings, this implicit learning fails because the training data lacks the coverage needed to set clear boundaries.

NSL-MT makes these boundaries explicit. By generating violations that target known error patterns, we provide negative evidence that would require orders of magnitude more parallel data to learn implicitly. A model trained on 15,000 positive examples might never see enough instances of correct adjective-noun order to reliably infer the rule. However, exposure to 60,000 explicit violations of adjective-noun order—4 violations per positive example—creates an unambiguous learning signal.

The severity weighting mechanism is important to NSL-MT effectiveness. Not all errors carry equal importance for communication. Gender agreement violations in languages without grammatical gender represent fundamental category errors that severely impair comprehension. Article insertion violations create awkward but generally comprehensible output. By weighting violations according to their impact on meaning, NSL-MT guides models to prioritize avoiding errors while tolerating minor stylistic deviations when necessary.

Language-Specific Effects The ablation study reveals that different constraint types contribute differently across languages. Zarma benefits most from lexical constraints because of its 'heavy' reliance on particles and auxiliaries to express grammatical relations. Bambara shows the largest gains from syntactic constraints because its 'rigid' SOV word order differs from French SVO patterns. Fulfulde demonstrates strong improvements from morphological constraints.

These patterns validate our theoretical motivation for NSL-MT. The method works best when it targets the specific structural differences between source and target languages.

Data Efficiency The results in terms of data efficiency reveal two distinct patterns of NSL-MT efficiency. At very low data sizes (100-500 examples), NSL-MT prevents complete training failure. Without NSL-MT, models produce incoherent output because they lack sufficient signal to identify basic patterns. NSL-MT provides structure that enables learning even from minimal positive examples.

At moderate data sizes (1,000-5,000 examples), NSL-MT accelerates convergence to solutions that normal training would eventually reach with more data. The 5x data efficiency multiplier we observe at 1,000 examples represents a practical threshold where NSL-MT makes previously infeasible projects viable. Collecting 5,000 parallel sentences might cost thousands of dollars, while creating 20 linguistic rules requires hours of native speaker consultation.

At high data sizes (15,000+ examples), NSL-MT continues improving performance but the relative advantage shrinks. This pattern suggests that NSL-MT primarily compensates for insufficient training data rather than fundamentally changing what models can learn. Given unlimited parallel data, normal

training might eventually match NSL-MT performance. However, unlimited parallel data remains infeasible—for now—for most low-resource languages.

Cross-Architecture Generalization NSL-MT improves all four tested architectures differences, and parameter sizes. NLLB-200 benefits least because its massive multilingual pre-training already captures many cross-lingual patterns. AfriMT5 benefits because its pre-training covers African languages but lacks the scale of NLLB. The mT5 variants benefit most from NSL-MT.

This generalization pattern indicates that NSL-MT effectiveness depends more on the linguistic properties of the translation task than on specific architectural choices. The method works by providing information that models need but cannot easily extract from positive examples alone and small training set. Any architecture that uses gradient-based learning to optimize likelihood will benefit from explicit negative constraints.

Implications NSL-MT enables MT development for languages where traditional approaches fail due to insufficient parallel data. The method requires two resources: a small parallel corpus and native speakers who can create grammatical rules. The first resource exists for hundreds of languages but are too small for effective training. The second resource exists for thousands of languages but remains underused by current methods.

The time investment for NSL-MT implementation remains modest. Creating violation generators for the three languages in our study required approximately 15 hours of linguist consultation plus 10 hours of programming (if no AI tool used). This investment produces reusable tools that apply to any translation model. In contrast, collecting an additional 10,000 parallel sentences would require weeks of translator time and cost thousands of dollars.

5 Conclusion

We introduce Negative Space Learning, a training method that teaches translation models what not to generate by augmenting standard parallel data with linguistically informed violations. NSL-MT addresses a fundamental limitation of maximum likelihood training in low-resource settings: models receive massive information about correct translations but no explicit information about incorrect translations. By generating negative examples that violate specific grammatical constraints, NSL-MT provides robust learning signal that would otherwise require orders of magnitude more parallel data.

Our experiments on French-to-African translation demonstrate that NSL-MT improves performance across different model architectures. NSL-MT provides particularly benefits in severely data-constrained scenarios, offering a 5x data efficiency multiplier at 1,000 training examples. The method proves most effective for constraint types that target the specific structural differences between source and target languages, confirming that linguistic knowledge encoded as negative constraints enables more efficient learning than distributional information alone.

6 Limitations

We acknowledge several limitations of our work that suggest directions for future research.

6.1 Violation Generator Quality

NSL-MT effectiveness depends on the quality of violation generators. Our generators encode linguistic knowledge obtained through grammar descriptions and native speaker consultation. However, this knowledge remains incomplete. We target common error patterns but cannot enumerate all possible violations of target language grammar. More comprehensive violation sets might improve NSL-MT performance further, but creating them requires deeper linguistic analysis.

Language Coverage We evaluate NSL-MT on only three languages. NSL-MT might prove less effective for languages more similar to their source languages. We also test only French-to-African translation. The method might work differently for other language pairs or translation directions.

Evaluation Scope We rely mainly on automatic metrics to evaluate translation quality. While BLEU, chrF++, and COMET correlate with human judgments, they do not capture all aspects of translation adequacy. COMET in particular uses learned representations that might not fully capture the semantic nuances of low-resource languages. Human evaluation would provide stronger evidence but remains expensive at scale.

Computational Cost NSL-MT increases training time by approximately 4x due to the additional

negative examples in each batch. This overhead remains manageable for research experiments but might pose challenges for large-scale cases. The violation generation process itself requires minimal computation but introduces engineering complexity.

Generalization Beyond Translation We demonstrate NSL-MT effectiveness for machine translation but do not evaluate its applicability to other natural language generation tasks. The core principle of learning from negative examples should transfer to tasks like text summarization, dialogue generation, or grammatical error correction. However, these tasks might require different violation strategies than those we use for translation.

References

David Ifeoluwa Adelani and 1 others. 2022. A few thousand translations go a long way! leveraging pre-trained models for african news translation. In *Proceedings of NAACL-HLT*, pages 3053–3070.

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. In *Proceedings of NAACL-HLT*, pages 3874–3884.

Verena Blaschke, Masha Fedzechkina, and Maartje ter Hoeve. 2025. Analyzing the effect of linguistic similarity on cross-lingual transfer: Tasks and experimental setups matter. *Preprint*, arXiv:2501.14491.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, pages 1597–1607. PMLR.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910.

Mamadou K. Keita, Christopher Homan, and Sebastien Diarra. 2025. R2t: Rule-encoded loss functions for low-resource sequence tagging. *Preprint*, arXiv:2510.13854.

Yu-Hsiang Lin, Chian-Yu Chen, Jean Lee, Zirui Li, Yuyan Zhang, Mengzhou Xia, Shruti Rijhwani, Junxian He, Zhisong Zhang, Xuezhe Ma, and 1 others. 2019. Choosing transfer languages for cross-lingual learning. In *Proceedings of ACL*, pages 3125–3135.

Alexandre Magueresse, Vincent Carles, and Evan Heetderks. 2020. Low-resource languages: A review of past work and future challenges. *Preprint*, arXiv:2006.07264.

Jonathan Mallinson, Rico Sennrich, and Mirella Lapata. 2017. Paraphrasing revisited with neural machine translation. In *Proceedings of EACL*, pages 881–893.

Long Ouyang, Jeffrey Wu, Xu Jiang, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Kishore Papineni and 1 others. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of ACL*, pages 311–318.

Maja Popović. 2017. chrf++: words helping character n-grams. In *Proceedings of WMT*, pages 612–618.

Syed Matla Ul Qumar, Muzaffar Azim, and S. M. K. Quadri. 2025. Enhancing low-resource neural machine translation with decoding-based data augmentation. *International Journal of Information Technology*.

Ricardo Rei and 1 others. 2020. Comet: A neural framework for mt evaluation. In *Proceedings of EMNLP*, pages 2685–2702.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of ACL*, pages 86–96.

NLLB Team and 1 others. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2022. Finetuned language models are zero-shot learners. *International Conference on Learning Representations*.

Linting Xue and 1 others. 2021. mt5: A massively multilingual pre-trained text-to-text transformer. *Proceedings of NAACL-HLT*, pages 483–498.

JiaJun Zhang and ChengQing Zong. 2020. Neural machine translation: Challenges, progress and future. *Science China Technological Sciences*, 63(10):2028–2050.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. In *Proceedings of EMNLP*, pages 1568–1575.

A Related Work

Low-resource machine translation has received growing attention as the field moves beyond high-resource language pairs. Early approaches focused on transfer learning from high-resource languages (Zoph et al., 2016) or multilingual training that shares representations across language families (Aharoni et al., 2019). These methods improve over

monolingual baselines but struggle when source and target languages has typological differences. Our work addresses this limitation by explicitly encoding cross-lingual structural differences as negative constraints.

Recent multilingual pre-trained models demonstrate strong cross-lingual transfer capabilities. mT5 (Xue et al., 2021) pre-trains on 101 languages using a masked span prediction objective, while NLLB (Team et al., 2022) trains on 200 languages using a combination of parallel data and back-translation. AfriMT5 (Adelani et al., 2022) specializes multilingual pre-training for African languages. These models provide strong starting points but still require fine-tuning data to achieve good performance on low-resource languages. We demonstrate that NSL-MT improves fine-tuning efficiency across different model architecture, suggesting that pre-training and explicit linguistic constraints provide dual benefits.

Data augmentation methods create synthetic training examples to increase effective dataset size. Back-translation generates source sentences from target monolingual data, while paraphrasing creates alternative translations of existing parallel sentences (Qumar et al., 2025; Mallinson et al., 2017; Sennrich et al., 2016). These approaches expand the training distribution but maintain standard maximum likelihood objectives. NSL-MT differs fundamentally by modifying the training objective itself to include negative evidence. Furthermore, back-translation requires strong reverse-direction models that do not exist for most low-resource languages, while NSL-MT requires only linguistic knowledge that native speakers can provide.

Contrastive learning frameworks have gained in representation learning and natural language processing (Gao et al., 2021; Chen et al., 2020). These methods learn by distinguishing positive examples from negative examples in embedding space. NSL-MT shares the core principle of learning from negative evidence but operates at the sequence level rather than the representation level. Standard contrastive methods generate negatives through random sampling or data augmentation, creating easy negatives that are far from decision boundaries. NSL-MT generates negatives through targeted linguistic violations, creating hard negatives that specifically address known model failure modes. This distinction proves useful for low-resource settings where random negatives provide insufficient learning signal.

Reinforcement learning from human feedback (RLHF) is considered as a powerful technique for aligning language models with human preferences (Ouyang et al., 2022). RLHF trains reward models on human judgments of output quality, then uses these rewards to fine-tune generation models. This approach shares NSL-MT's goal of teaching models what not to generate. However, RLHF requires collecting human judgments at scale, which is expensive for low-resource languages. NSL-MT provides a practical alternative by encoding linguistic constraints that would be expensive to learn from human feedback. Where RLHF learns from implicit human preferences, NSL-MT encodes explicit linguistic rules.

Work on cross-lingual transfer has investigated how linguistic typology affects transfer success. Studies show that syntactic similarity predicts transfer effectiveness better than genetic relatedness or geographical proximity (Blaschke et al., 2025; Lin et al., 2019). This finding aligns our constraint type ablation study, which reveals that different violation categories contribute differently across languages based on their typological properties. Our results confirm that effective transfer requires explicit attention to structural differences between source and target languages.

Instruction tuning research has shown that task descriptions and demonstrations improve model performance on downstream tasks (Wei et al., 2022). This work demonstrates the value of explicit task specification beyond implicit pattern learning. NSL-MT applies similar principles to translation quality by explicitly specifying what constitutes incorrect output. While instruction tuning tells models what to do, NSL-MT tells models what not to also do, providing complementary forms of explicit guidance.

B Hyperparameter Analysis

We conduct additional experiments to assess NSL-MT robustness to hyperparameter choices. These experiments use Zarma language with AfriMT5-base trained on 5,000 examples for 3 epochs. We investigate two key hyperparameters: the negative weight α and the violation ratio.

B.1 Alpha Sensitivity

The negative weight hyperparameter α in Equation 2 controls the importance of negative examples in the training objective. We test four values:

$\alpha \in \{0.3, 0.5, 0.7, 0.9\}$ to determine NSL-MT sensitivity to this parameter.

Table 4 presents the results. NSL-MT performance remains stable across different α values, with BLEU scores varying by only 0.54 points (27.13-27.67) and chrF++ scores varying by 0.21 points (43.87-44.08). The COMET scores show similar stability, ranging from 0.7512 to 0.7535. This performance demonstrates that NSL-MT effectiveness does not necessarily depend on precise α tuning.

The descent advantage of lower α values (0.3-0.5) over higher values (0.9) suggests that overly aggressive penalization of negative examples can slightly degrade performance. However, the differences remain within confidence intervals, indicating that any $\alpha \in [0.3, 0.9]$ produces competitive results. We selected $\alpha = 0.7$ for our main experiments based on validation set performance, but these results show that alternative choices would produce comparable outcomes.

Alpha (α)	BLEU	chrF++	COMET
0.3	**27.67**	**44.03**	**0.7535**
0.5	27.50	43.97	0.7528
0.7	27.64	44.08	0.7512
0.9	27.13	43.87	0.7523
Range	0.54	0.21	0.0023

Table 4: Alpha sensitivity analysis on Zarma using AfriMT5-base with 5,000 training examples. Performance remains stable across different α values, varying by less than 2% across all metrics.

B.2 Violation Ratio Sensitivity

The violation ratio determines how many negative examples go with each positive example during training. We test three ratios by varying the number of violations generated per positive example: 2:1 (generating 1-2 violations), 4:1 (generating 3-5 violations, our default), and 6:1 (generating 5-7 violations).

Table 5 shows that violation ratio significantly impacts NSL-MT effectiveness. The 2:1 ratio produces lower scores (19.44 BLEU, 35.52 chrF++, 0.7188 COMET), suggesting insufficient negative signal for effective learning. The 4:1 ratio yields moderate performance (27.11 BLEU, 43.71 chrF++, 0.7512 COMET), while the 6:1 ratio achieves the best results (30.69 BLEU, 47.64 chrF++, 0.7689 COMET).

These results reveal that NSL-MT benefits from higher violation ratios, with the 6:1 ratio improving BLEU by 13.1% over the 4:1 default and by 57.8% over the 2:1 ratio. This suggests that exposing models to more diverse negative examples strengthens their ability to distinguish valid from invalid outputs. However, we note that ***higher ratios increase computational cost proportionally***. The 4:1 ratio represents a practical balance between effectiveness and efficiency, though researchers with sufficient computational resources may benefit from using 6:1 or higher ratios.

The gap between 2:1 and higher ratios indicates that NSL-MT requires a minimum threshold of negative examples to function effectively. With too few violations, the model may not encounter sufficient coverage of error patterns, limiting NSL-MT's ability to create clear grammatical boundaries.

Violation Ratio	BLEU	chrF++	COMET
2:1	19.44	35.52	0.7188
4:1 (default)	27.11	43.71	0.7512
6:1	**30.69**	**47.64**	**0.7689**

Table 5: Violation ratio sensitivity analysis on Zarma using AfriMT5-base with 5,000 training examples. Higher violation ratios provide stronger learning signal, with 6:1 outperforming 4:1 by 13.1% BLEU and 2:1 by 57.8% BLEU.

B.3 Discussion

These experiments demonstrate that NSL-MT deliver desirable robustness properties. The α hyperparameter shows minimal sensitivity, allowing anyone to select values in the range [0.3, 0.9] without careful tuning. This robustness simplifies NSL-MT usage for new language pairs where validation data may be limited.

In contrast, the violation ratio requires more careful consideration. Our results suggest using ratios of at least 4:1, with 6:1 or higher providing additional benefits when computational resources permit. The strong performance gains from higher ratios validate NSL-MT's core principle: explicit negative evidence accelerates learning, and more negative evidence yields stronger improvements.

We also observe that these findings support our main results. The 4:1 ratio used in our primary experiments represents a conservative choice that balances effectiveness and efficiency. The even stronger results at 6:1 suggest that our reported improvements may 'underestimate' NSL-MT's full potential when computational constraints are man-

ageable.