

A Lightweight Dual-Channel System for WMT26 Chinese-Southeast Asian Machine Translation

Xiaojing Yan and Rile Hu

Beijing Babel Technology, Co., Ltd.

hurile@aibabel.com

Abstract

We describe a lightweight system for the WMT26 Chinese-Southeast Asian Multilingual Translation Task, covering 14 directions between Chinese and seven Southeast Asian languages. It uses two independently trained Transformer Base models: a one-to-many (OTM) model for translation from Chinese and a many-to-one (MTO) model for translation into Chinese. Each model has 68,727,808 unique parameters. Both use 134,645 real parallel sentence pairs, augmented with 154,452 back-translated pairs for OTM and a 96,000-pair subset with synthetic Chinese targets for MTO. On the 6,859-pair internal test set, the final CTranslate2-based system achieves direction-family macro-average BLEU scores of 43.70 and 49.03; on the FLORES-200 cross-domain test set, they are 17.52 and 26.03, respectively. A CT2-only microbenchmark on a single NVIDIA GeForce RTX 5090 achieves 237.02 and 241.96 sentences per second on presegmented, source-length-stratified development samples, using the median of three steady-state measurements.

1 Introduction

Machine translation between Chinese and Southeast Asian languages must handle diverse writing systems with limited parallel supervision. The WMT26 Chinese-Southeast Asian Multilingual Translation Task covers both translation directions between Chinese and Indonesian, Khmer, Lao, Malay, Burmese, Thai, and Vietnamese. Its evaluation considers translation quality and inference throughput, making deployment cost an explicit part of system design.¹

Multilingual parameter sharing can transfer useful information across languages, but its effect depends on model capacity, data availability, and task sampling (Aharoni et al., 2019; Shaham et al.,

2023). The Chinese-centered structure of this task provides a natural intermediate level of sharing. Translation from Chinese requires multilingual generation, whereas translation into Chinese shares a single target language. We organize the 14 directions into these two families rather than training one model per language pair or requiring all directions to share a single complete model.

Our system uses an independently trained one-to-many (OTM) Transformer and many-to-one (MTO) Transformer. Development included exploration of unified many-to-many (MTM) and direction-grouped models, leading to the final two-channel design. At inference time, a supplied translation-direction argument determines which model is called.

The direction split also exposes an important difference in synthetic supervision. A Southeast Asian monolingual sentence paired with a teacher-generated Chinese translation supplies a synthetic source and a real target to OTM, but a real source and a synthetic target to MTO. We therefore use different synthetic-data volumes for the two channels: 154,452 and 96,000 sentence pairs, respectively. These configurations allow the two channels to use different synthetic-target shares.

We describe the system architecture and asymmetric data augmentation, and evaluate the translation quality and decoding throughput of the final CTranslate2-based system.

2 Related Work

Multilingual sharing and specialization. Language tags, shared vocabularies, and shared model parameters are standard tools for multilingual neural machine translation (Johnson et al., 2017). Many-to-many models can support transfer across numerous translation directions (Aharoni et al., 2019), while partial sharing offers other trade-offs between multilingual representations and task-

¹<https://www2.statmt.org/wmt26/chinese-south-east-asian.html>

specific components (Firat et al., 2016). Interference depends on capacity, data size, and language-pair proportions, rather than on sharing alone (Shaham et al., 2023). In contrast to multilingual denoising pretraining (Liu et al., 2020), we train compact Transformers from scratch and share parameters within two direction families.

Monolingual and synthetic supervision. Back-translation pairs target-language monolingual text with synthetic source sentences (Sennrich et al., 2016). Its utility depends on the generation method and training configuration (Edunov et al., 2018). Translating source-side monolingual text instead supplies synthetic targets (Zhang and Zong, 2016); multilingual systems have used both forms of augmentation (Xie et al., 2021). Our contribution is a system configuration that reuses a common pool of teacher translations in opposite directions with different retained volumes, rather than a new synthetic-data generation method.

Efficient translation. Efficiency-oriented shared tasks emphasize the joint reporting of quality, model size, and measured speed under specified execution conditions (Heafield et al., 2022). We train Transformers (Vaswani et al., 2017) with fairseq (Ott et al., 2019) and deploy them through CTranslate2.²

3 Task Data and Evaluation Sets

3.1 Training Data

Let $\mathcal{L} = \{\text{id, km, lo, ms, my, th, vi}\}$ denote Indonesian, Khmer, Lao, Malay, Burmese, Thai, and Vietnamese, respectively; Chinese is denoted by zh. The task comprises $\text{zh} \leftrightarrow \ell$ for every $\ell \in \mathcal{L}$, giving 14 translation directions. The organizers impose a 20-billion-parameter limit on submitted systems.

The official training resources contain 140,000 parallel sentence pairs, with 20,000 per language pair, and 700,000 Southeast Asian monolingual sentences, with 100,000 per language. Parallel-data cleaning includes format and empty-field checks, removal of anomalous markers, length and length-ratio filtering, exact deduplication, and language/script consistency checks. The resulting real parallel corpus contains 134,645 sentence pairs. Both channels use this same corpus, with the source and target sides reversed as appropriate.

Monolingual cleaning retains 683,416 sentences. From this pool, we construct 165,400 teacher-

Stage	Input	Retained
Parallel cleaning	140,000	134,645
Monolingual cleaning	700,000	683,416
Teacher-candidate filtering	165,400	154,452
MTO subset selection	154,452	96,000

Table 1: Training-data processing stages.

translation candidates and retain 154,452 synthetic pairs after filtering. OTM uses all retained pairs; MTO uses a 96,000-pair subset. Table 1 summarizes the processing stages; monolingual counts are sentences, and all other counts are sentence pairs. Candidate selection is distinct from initial monolingual cleaning. Appendix A gives the per-language counts.

3.2 Internal Development and Test Splits

The official development resource contains 14,000 parallel pairs, with 2,000 per language pair. Within each language pair, we randomly divide it into two equal parts before cleaning: 1,000 pairs for the internal development set (Dev-in) and 1,000 for the internal test set (Test-in). The split unit is a parallel pair, and the two translation directions use the same assignment.

After cleaning, Dev-in contains 6,862 pairs, or 13,724 directional examples when used bidirectionally. We use Dev-in for training monitoring and for selecting models, training duration, checkpoints, and decoding parameters, but not as gradient-update supervision.

The cleaned Test-in contains 6,859 pairs, or 13,718 directional examples. It is reserved for final quality evaluation and is excluded from model, checkpoint, and decoding-parameter selection. Thus, Test-in is the held-out half of the official development resource after cleaning, not the official blind-test set. Table 5 provides the full split counts.

3.3 Cross-Domain Test Set

We use the official dev split of FLORES-200 as a cross-domain test set, with 997 sentences per language (13,958 directional examples).³ We refer to this set as FLORES. It is used only for final quality evaluation, after training duration, model and checkpoint choices, and decoding parameters have been fixed. Neither FLORES nor Test-in informs these choices. Official blind-test evaluation

²<https://opennmt.net/CTranslate2/>

³<https://huggingface.co/datasets/facebook/flores>

remains separate; we report neither official scores nor an official throughput ranking.

4 Direction Grouping and Asymmetric Augmentation

4.1 Two Independently Trained Channels

As illustrated in Figure 1, the system uses independently trained OTM and MTO models for the two translation-direction families. The OTM model represents $p_{\theta_o}(y_\ell | x_{zh}, \ell)$. It encodes Chinese and generates one of seven Southeast Asian languages, conditioned on a target-language tag. The MTO model represents $p_{\theta_m}(y_{zh} | x_\ell, \ell)$, sharing its encoder across the seven source languages and using a Chinese decoder. The parameter sets θ_o and θ_m are independent. Each translation uses only the model specified by the direction argument, without combining the two models’ output probabilities.

Both channels use Transformer Base with six encoder layers, six decoder layers, a hidden dimension of 512, a feed-forward dimension of 2,048, and eight attention heads. We use pre-layer normalization in the encoder and decoder and layer normalization on embeddings. The decoder input embedding and output projection share the same parameter tensor. The encoder input embedding is separate.

Each model has 68,727,808 unique parameters, counted by summing the sizes of the distinct model parameters. Swapping the 16,009-entry Chinese vocabulary and the 32,009-entry Southeast Asian vocabulary between source and target leaves the combined embedding size unchanged. Thus the two channels have equal parameter counts despite different target vocabularies. The teacher used for offline data generation is not part of the online model.

4.2 Tokenization and Language Conditioning

We train separate unigram SentencePiece models (Kudo and Richardson, 2018): 16,000 pieces for Chinese and 32,000 shared pieces for the seven Southeast Asian languages. Their training data consist of real and synthetic training text. Runtime vocabularies contain 16,009 and 32,009 entries after including the required special symbols and language tags. OTM uses the Chinese vocabulary on the source side and the Southeast Asian vocabulary on the target side; MTO reverses them. A target-language tag initiates decoding.

Training samples directions uniformly within

each channel. This equalizes direction-selection probabilities, not token counts or the number of times each sentence is observed. Sentence lengths, vocabulary segmentation, and per-language data volumes still affect training exposure.

4.3 The Two Roles of Synthetic Pairs

Let s_ℓ be an authentic Southeast Asian monolingual sentence and let $\hat{z} = T(s_\ell)$ be its teacher-generated Chinese translation. The two synthetic training sets have the forms

$$\begin{aligned}\mathcal{D}_o^{\text{syn}} &= \{(\hat{z}, s_\ell)\}, \\ \mathcal{D}_m^{\text{syn}} &= \{(s_\ell, \hat{z})\}.\end{aligned}\tag{1}$$

For OTM, the real sentence is the target, so these pairs provide back-translation supervision. For MTO, the Chinese target is synthetic, so they provide forward-translation-style pseudo-labels. Since the teacher and student are different models, we do not describe the latter as student self-training.

We use google/gemma-4-31B-it as the offline teacher.⁴ Adding the retained synthetic data to the real corpus gives 289,097 training pairs for OTM and 230,645 for MTO. The MTO subset is selected by strata defined by language, corpus domain, and Southeast Asian SentencePiece length decile, using largest-remainder allocation for quotas.

For channel k , define the sentence-pair synthetic share as

$$\rho_k = \frac{|\mathcal{D}_k^{\text{syn}}|}{|\mathcal{D}_k^{\text{real}}| + |\mathcal{D}_k^{\text{syn}}|}.\tag{2}$$

The final shares are 53.43% for OTM and 41.62% for MTO. They describe corpus composition, not separate loss weights or token-level sampling probabilities.

This asymmetry limits teacher-generated Chinese targets to 41.62% of the MTO training pairs, rather than reusing the full synthetic pool. Teacher errors enter the source side for OTM but directly affect target supervision for MTO, motivating separate data-volume choices.

4.4 Optimization and Checkpoint Averaging

We train from scratch using label-smoothed cross-entropy with smoothing 0.1, Adam, a peak learning rate of 5×10^{-4} , 4,000 warmup updates, and inverse-square-root decay. Each microbatch contains at most 8,192 tokens, with gradient accumulation over five microbatches. Training uses FP16 on a single RTX 5090 with fairseq 0.12.3 (local

⁴<https://huggingface.co/google/gemma-4-31B-it>

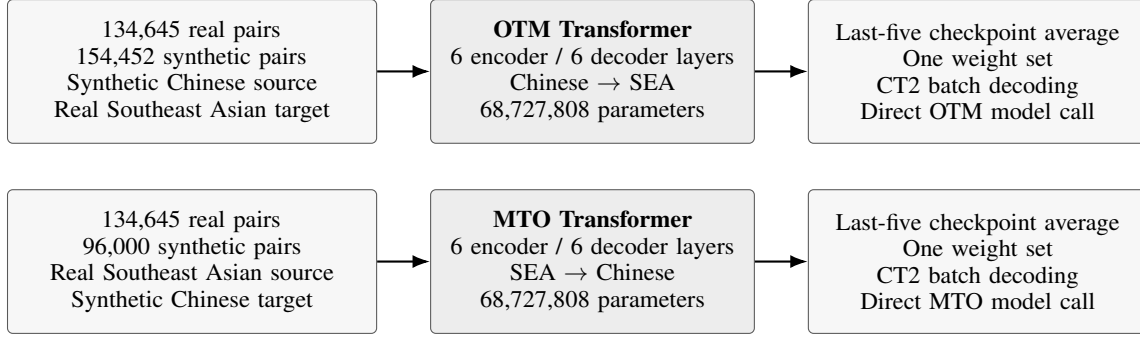


Figure 1: OTM and MTO training and deployment.

build). We use Dev-in for validation and training-duration and checkpoint selection, and save checkpoints every 5,000 updates. Additional training settings appear in Appendix B.

The final training endpoints are 225,000 updates for OTM and 220,000 for MTO. For each channel, we take the elementwise arithmetic mean of the final five saved checkpoints along that training trajectory. This yields one set of weights per channel, not an inference-time ensemble. The averaged updates are 205k to 225k for OTM and 200k to 220k for MTO, in 5k increments. We convert the averaged checkpoints to CTranslate2 for translation generation and deployment.

5 Translation Quality

5.1 Evaluation Protocol

We evaluate on the held-out Test-in and FLORES test sets described in Section 3. All model, checkpoint, and decoding settings are fixed using Dev-in before test evaluation.

We report corpus BLEU, chrF2, and COMET (Post, 2018; Popović, 2015; Rei et al., 2020). BLEU and chrF2 use SacreBLEU 2.6.0. BLEU uses one reference, mixed case, exponential smoothing, and eff: no, with the Chinese tokenizer (tok: zh) for Chinese targets and the FLORES-200 tokenizer (tok: flores200) for Southeast Asian targets. chrF2 uses character 6-grams, $\beta = 2$, no word n-grams, and excludes whitespace; the BLEU tokenizer choice does not apply to chrF2.

COMET uses unbabel-comet 2.2.7 and the Unbabel/wmt22-comet-da model (Rei et al., 2022). We retain the model’s raw score scale, without multiplication by 100. We use a fixed evaluator configuration because learned-metric scores depend on the evaluation setup (Zouhar et al., 2024).

Each metric is computed separately for each direction. We then average the seven directional

scores within each family $\mathcal{F}$:

$$\overline{M}_{\mathcal{F}} = \frac{1}{7} \sum_{d \in \mathcal{F}} M_d. \quad (3)$$

This is an unweighted direction-level macro-average, not a score on concatenated multilingual outputs. Because OTM and MTO have different target languages, their absolute metric values should not be interpreted as a direct comparison of the two families’ intrinsic difficulty or model quality.

Table 2 reports translation quality obtained from the final CTranslate2-based system using the averaged OTM and MTO models. For each direction, BLEU, chrF2, and COMET are computed from the same translations.

5.2 Direction-Level Results

OTM and MTO achieve macro-average BLEU scores of 43.70 and 49.03 on Test-in, and 17.52 and 26.03 on FLORES. All three family-level metric averages are lower on FLORES than on Test-in. These differences characterize a cross-dataset quality gap.

Performance is uneven across directions. For OTM, Test-in BLEU ranges from 33.45 for zh-my to 56.48 for zh-vi. On FLORES, zh-my also has the lowest OTM BLEU, at 10.56. In the reverse family, my-zh has the lowest BLEU on both test sets. These patterns identify Burmese-related directions as priorities for data-coverage and error analysis, rather than allowing the macro-average to conceal weaker directions.

Metric rankings also differ. The zh-my direction has the lowest Test-in BLEU within OTM, but the highest COMET score, 0.903. This disagreement alone cannot determine whether the cause is lexical mismatch, reference variation, or a limitation of the learned evaluator. We therefore retain all three metrics and the complete per-direction table

Model	Direction	Test-in			FLORES-200 (test)		
		BLEU	chrF2	COMET	BLEU	chrF2	COMET
OTM	zh-id	48.25	70.01	0.853	18.81	49.04	0.806
OTM	zh-km	38.64	58.28	0.863	13.08	39.86	0.776
OTM	zh-lo	35.96	61.52	0.856	14.57	38.86	0.763
OTM	zh-ms	49.80	72.29	0.863	18.87	49.37	0.802
OTM	zh-my	33.45	58.46	0.903	10.56	32.66	0.759
OTM	zh-th	43.31	60.61	0.845	25.52	41.92	0.804
OTM	zh-vi	56.48	70.65	0.898	21.26	42.87	0.800
OTM	Macro avg.	43.70	64.55	0.869	17.52	42.08	0.787
MTO	id-zh	57.28	57.76	0.833	28.64	27.12	0.787
MTO	km-zh	46.39	46.16	0.838	25.16	24.70	0.790
MTO	lo-zh	42.53	41.50	0.842	25.52	25.35	0.794
MTO	ms-zh	57.87	54.01	0.838	27.92	26.74	0.774
MTO	my-zh	40.26	37.09	0.851	21.40	20.53	0.766
MTO	th-zh	43.12	41.40	0.818	26.51	25.29	0.813
MTO	vi-zh	55.78	50.90	0.864	27.04	26.03	0.793
MTO	Macro avg.	49.03	46.97	0.841	26.03	25.11	0.788

Table 2: Translation quality of the final CTranslate2-based system on two test sets. Test-in contains 6,859 pairs from the held-out 50% of the official development resource after cleaning. FLORES uses the official FLORES-200 dev split (997 sentences per language) exclusively for final testing. Neither set is used for model selection or tuning. Macro-averages are computed over seven directions before rounding.

instead of equating a high score on one metric with demonstrated human-perceived superiority.

5.3 Interpretation of the Final Configuration

The reported scores characterize the complete system configuration, combining direction grouping, asymmetric augmentation, channel-specific training, and checkpoint averaging. The supervision roles in Equation 1 motivate the data design, and Table 2 reports the resulting translation quality. The evaluation characterizes the combined configuration rather than individual component effects.

6 Deployment and CT2-Only Throughput

6.1 Single-Channel Inference

The deployment system uses CTranslate2 4.8.1 and SentencePiece 0.2.1. The caller supplies a translation-direction argument, such as zh-id for OTM or id-zh for MTO, and directly invokes the corresponding model. No separate routing module is used. The same argument specifies the vocabularies, target-language tag, and decoding settings. The GPU path uses FP16, a batch limit of 128 examples, and beam size 4. Direction-specific deployment settings appear in Table 8.

Each request invokes one 68,727,808-parameter model. Checkpoint averaging introduces no additional decoding branches, and the teacher is absent from online inference. Appendix C.2 documents device-level memory monitoring.

Channel	Unique parameters	Sentences/s
OTM	68,727,808	237.02
MTO	68,727,808	241.96

Table 3: CT2-only throughput on sampled Dev-in inputs (median of three measurements).

6.2 Microbenchmark Protocol

We measure the two channels separately on one NVIDIA GeForce RTX 5090 using CT2 FP16, batches of 128 examples, and beam size 4. For each channel, a measurement uses 128 Dev-in sentences selected by source-length quantiles. The input has already been segmented with SentencePiece. We warm up on 16 examples and then perform three steady-state measurements.

For channel k , throughput is the median of the three completed-sentence-count-to-inference-time ratios:

$$T_k = \text{median}_{r \in \{1,2,3\}} \frac{N_{k,r}}{t_{k,r}^{\text{CT2}}}, \quad (4)$$

where $N_{k,r} = 128$ and $t_{k,r}^{\text{CT2}}$ is the CT2 inference time for that measurement. There is no cross-direction weighting or aggregation of per-direction throughputs. Table 3 reports the results.

Only the CT2 inference interval enters the denominator. Model loading is recorded separately and excluded from steady-state timing, as is warmup. SentencePiece processing is outside

the timed path. Long-sentence splitting and exceptional redecoding are disabled. These figures therefore characterize CT2-only decoding on the sampled development inputs, not SentencePiece-plus-CT2 or end-to-end service throughput.

7 Conclusion

We present a compact dual-channel system for Chinese-Southeast Asian translation. Two independently trained Transformers share parameters within opposite direction families and use different volumes of synthetic supervision. Each model contains 68,727,808 unique parameters. The final CTranslate2-based system obtains OTM/MTO macro-average BLEU scores of 43.70/49.03 on Test-in and 17.52/26.03 on the FLORES-200 test set. A single-GPU, CT2-only microbenchmark yields 237.02/241.96 sentences per second under the specified steady-state protocol. The system combines direction-family parameter sharing, asymmetric synthetic supervision, and single-model decoding for compact multilingual translation.

References

- Roei Aharoni, Melvin Johnson, and Orhan Firat. 2019. [Massively multilingual neural machine translation](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3874–3884, Minneapolis, Minnesota. Association for Computational Linguistics.
- Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. [Understanding back-translation at scale](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.
- Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016. [Multi-way, multilingual neural machine translation with a shared attention mechanism](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 866–875, San Diego, California. Association for Computational Linguistics.
- Kenneth Heafield, Biao Zhang, Graeme Nail, Jelmer Van Der Linde, and Nikolay Bogoychev. 2022. [Findings of the WMT 2022 shared task on efficient translation](#). In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 100–108, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.
- Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. [Google’s multilingual neural machine translation system: Enabling zero-shot translation](#). *Transactions of the Association for Computational Linguistics*, 5:339–351.
- Taku Kudo and John Richardson. 2018. [SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.
- Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. [Multilingual denoising pre-training for neural machine translation](#). *Transactions of the Association for Computational Linguistics*, 8:726–742.
- Myle Ott, 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. Association for Computational Linguistics.
- Maja Popović. 2015. [chrF: character n-gram F-score for automatic MT evaluation](#). In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.
- Matt Post. 2018. [A call for clarity in reporting BLEU scores](#). In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.
- Ricardo Rei, José G. C. de Souza, Duarte Alves, Chrysoula Zerva, Ana C Farinha, Taisiya Glushkova, Alon Lavie, Luisa Coheur, and André F. T. Martins. 2022. [COMET-22: Unbabel-IST 2022 submission for the metrics shared task](#). In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 578–585, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.
- Ricardo Rei, Craig Stewart, Ana C. Farinha, and Alon Lavie. 2020. [COMET: A neural framework for MT evaluation](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. [Improving neural machine translation models with monolingual data](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96,

Berlin, Germany. Association for Computational Linguistics.

Uri Shaham, Maha Elbayad, Vedanuj Goswami, Omer Levy, and Shruti Bhosale. 2023. [Causes and cures for interference in multilingual translation](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15849–15863, Toronto, Canada. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008. Curran Associates, Inc.

Wanying Xie, Bojie Hu, Han Yang, Dong Yu, and Qi Ju. 2021. [TenTrans large-scale multilingual machine translation system for WMT21](#). In *Proceedings of the Sixth Conference on Machine Translation*, pages 439–445, Online. Association for Computational Linguistics.

Jiajun Zhang and Chengqing Zong. 2016. [Exploiting source-side monolingual data in neural machine translation](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545, Austin, Texas. Association for Computational Linguistics.

Vilém Zouhar, Pinzhen Chen, Tsz Kin Lam, Nikita Moghe, and Barry Haddow. 2024. [Pitfalls and outlooks in using COMET](#). In *Proceedings of the Ninth Conference on Machine Translation*, pages 1272–1288, Miami, Florida, USA. Association for Computational Linguistics.

A Per-Language Data Counts

Tables 4 and 5 give the per-language breakdowns for the corpora described in Section 3. All counts are sentence pairs.

Language	Real	OTM synth.	MTO synth.
id	18,154	21,153	13,148
km	19,064	22,478	13,971
lo	19,379	22,811	14,178
ms	19,525	21,388	13,294
my	19,767	22,254	13,832
th	19,138	21,595	13,422
vi	19,618	22,773	14,155
Total	134,645	154,452	96,000

Table 4: Training sentence pairs by Southeast Asian language.

Pair	Official dev	Dev-in	Test-in
zh-id	2,000	886	891
zh-km	2,000	998	997
zh-lo	2,000	996	993
zh-ms	2,000	998	992
zh-my	2,000	997	1,000
zh-th	2,000	990	990
zh-vi	2,000	997	996
Total	14,000	6,862	6,859

Table 5: Development and test pairs after splitting and cleaning the official development resource. Dev-in is used for training monitoring and tuning; Test-in is used for final quality evaluation.

B Additional Training Settings

Table 6 supplements the architecture and optimization settings in Section 4. Table 7 gives random seeds and epoch/update limits; training stops when either limit is reached.

Setting	Value
Training task	translation_multi_simple_epoch
Tokenizer options	Character coverage 1.0; byte fallback enabled
SPM special IDs	UNK/BOS/EOS = 0/1/2; no built-in PAD
Adam settings	$\beta = (0.9, 0.98)$; $\epsilon = 10^{-6}$; weight decay 0
Dropout	Model 0.1; attention 0.1; activation 0.0
Position limit	1,024 for encoder and decoder

Table 6: Additional shared training settings.

Setting	OTM	MTO
Random seed	111	222
Epoch limit	90	200
Update limit	230,000	220,000
Final epoch	90	113

Table 7: Channel-specific seeds and training limits.

C Additional Runtime Details

C.1 Direction-Specific Decoding

Table 8 gives the length penalty (LP) and blocked repeated n-gram size (NR); NR=0 disables the repetition constraint. The translation interface preserves one output line per input line and the original input order.

OTM	LP	NR	MTO	LP	NR
zh-id	1.0	3	id-zh	1.0	0
zh-km	1.0	3	km-zh	0.8	0
zh-lo	1.0	3	lo-zh	1.0	3
zh-ms	1.0	3	ms-zh	1.2	0
zh-my	1.0	3	my-zh	1.0	3
zh-th	1.0	3	th-zh	1.0	0
zh-vi	1.2	0	vi-zh	1.0	0

Table 8: Direction-specific deployment settings.

C.2 Device-Memory Monitoring

A separate device-wide trace sampled with nvidia-smi every 100 ms records a 2,009 MiB baseline, a 3,127 MiB sampled peak (an increment of 1,118 MiB), and a 2,472 MiB final sample. The peak occurs during zh-km inference with OTM. Peak GPU utilization is 96%, not a memory-use percentage. These are session-level observations, not isolated model allocations or statistics over the three throughput repetitions.