Reference-Grounded Data Curation for Instruction-Following Thai-English Machine Translation

Thodsaporn Chay-intr[1,2], **Krittapad Harnchang**[1*], **Mahannop Thabua**[1*]
Kobkrit Viriyayudhakorn[1,3], **Thanaruk Theeramunkong**[2,4]
[1]iApp Technology, Thailand
[2]Intelligent Informatics and Service Innovation Research Center, Thailand
[3]Artificial Intelligence Entrepreneur Association of Thailand (AIEAT), Thailand
[4]Sirindhorn International Institute of Technology, Thammasat University, Thailand
{t.chayintr, krittapadpor12348, peemmygg}@gmail.com
kobkrit@aieat.or.th, thanaruk@siit.tu.ac.th

Abstract

Instruction-following machine translation (IF-MT) requires respecting prompt-level rules on terminology, formatting, and register. Rule compliance typically trades off against translation quality, a tension that general-purpose IF data augmentation methods do not address. We propose **Reference-Grounded Data Curation**, a two-phase pipeline that extracts every supervised constraint from a reference translation that already satisfies it, ensuring feasibility by construction. Phase 1 applies Instruction-Following Difficulty (IFD) scoring to retain the hardest-but-learnable instances from an English-Thai parallel pool. Phase 2 extracts constraints from each reference target and keeps only generations satisfying every constraint, yielding the 1.97M-record Grounded dataset. We fine-tune open-weight bases on Grounded to produce **ChindaMT**, a Thai-English translation family at 4B, 2B, and 0.8B parameters. Under length-controlled pairwise judging, ChindaMT outperforms or matches every same-size baseline at every tier on both plain translation and under explicit rules, reaching up to a 68.4% win rate against the strongest baseline. The recipe transfers cleanly across Qwen generations. We release model weights, the Grounded dataset, and evaluation suites.

1 Introduction

Large language models (LLMs) now translate competitively with dedicated MT systems (Xu et al., 2024b; Alves et al., 2024), and translation requests increasingly carry prompt-level rules on terminology, formatting, register, and length that outputs must satisfy while staying accurate. Yet translation-specialized fine-tuning erodes prompt-following capability, while instruction-tuned LLMs without translation specialization follow such rules but underperform on quality. Open-weight Thai-English systems illustrate both failure modes. Thai-focused LLMs such as Typhoon (Pipatanakul et al., 2023, 2024) support instruction following (IF) but lack specialized translation quality, while multilingual machine translation (MT) specialists such as Hunyuan-MT (Zheng et al., 2025) deliver translation quality without modeling prompt-level constraints. No open-weight Thai-English model currently closes this gap.

Two complementary lines of work build general IF training data. Data selection methods (Zhou et al., 2023a; Li et al., 2024b; Liu et al., 2024) identify a small high-quality SFT subset, while data augmentation methods (Wang et al., 2023; Dong et al., 2025; An et al., 2025) synthesize constraint-bearing instructions by prompting an LLM. Both target general IF benchmarks such as IFEval (Zhou et al., 2023b) and FollowBench (Jiang et al., 2024), but neither has been adapted to translation.

We propose **Reference-Grounded Data Curation** (RGDC), a two-phase pipeline that grounds every supervised constraint in an existing reference translation. Phase 1 applies the Instruction-Following Difficulty (IFD) score (Li et al., 2024b) to keep the hardest-but-learnable subset of an English-Thai parallel pool. Phase 2 extracts verifiable constraints from each retained reference, regenerates outputs, and keeps only candidates that pass every constraint under LLM-as-judge evaluation, producing the Grounded dataset.

We fine-tune four open-weight base models on Grounded to produce **ChindaMT**[1], a Thai-English translation family at 4B, 2B, and 0.8B parameters. Three variants use Qwen3.5 bases and a fourth uses

*Work done while at iApp Technology.

[1]Models: huggingface.co/iapp/ChindaMT-{4B,2B,0.8B}. Grounded and the suites: huggingface.co/datasets/iapp/ChindaMT-{Grounded,CoreEval,BroadEval}. Code: github.com/iapp-technology/ChindaMT-RGDC.

Proceedings of the 15th International Joint Conference on Natural Language Processing and the 5th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 1152–1171
November 6-10, 2026 ©2026 Association for Computational Linguistics

Qwen3-4B to demonstrate cross-generation transfer. Evaluation uses length-controlled pairwise win rate against same-size baselines on two suites we construct, with one primary and two open-weight cross-judges from independent model families. **CoreEval** covers five deployment-relevant domains spanning government, finance, tech, UI, and miscellaneous content, and **BroadEval** spans ten broader domains for cross-domain generalization. Each suite has a Plain split for translation only and a Constrained split with one to four rules.

On CoreEval, ChindaMT outperforms the size-matched MT specialists at all three sizes on both plain and constrained translation, and the same recipe transfers cleanly to Qwen3-4B as a cross-generation check. These wins extend to BroadEval and are corroborated by external metrics, two open-weight cross-judges from independent model families, and native-Thai human raters.

Our contributions are as follows:

- We propose **Reference-Grounded Data Curation**, which eliminates the compliance-quality conflict by extracting every constraint from a reference already satisfying it, so each rule-compliant target is a faithful translation.
- We release **ChindaMT**, an open-weight Thai-English translation family at 4B, 2B, and 0.8B parameters, with the Grounded dataset and the **CoreEval** and **BroadEval** suites.
- ChindaMT outperforms or matches every same-size baseline on plain and instruction-following translation, with external metrics, cross-judges, and humans agreeing and ablations isolating RGDC components.

2 Related Work

2.1 Data Selection for Instruction Tuning

LIMA (Zhou et al., 2023a) showed that small high-quality SFT data can match much larger uncurated sets (Chen et al., 2024; Luo et al., 2025), motivating data-selection methods that retain high-signal instances. The IFD score (Li et al., 2024b) selects samples where the instruction reduces but does not eliminate the loss on the output, yielding hard-but-learnable instances. Related methods include Deita (Liu et al., 2024), Humpback (Li et al., 2024c), and SuperFiltering (Li et al., 2024a).

We adapt IFD scoring to the translation setting because it defines difficulty with the loss of the base model being fine-tuned, so difficulty is measured for the model that will learn from the data, and needs no further scorer. SuperFiltering approximates this criterion with a weaker model to save cost. Deita scores complexity and quality, which have more room to separate samples in diverse open-domain pools than in our single-task pool.

2.2 Instruction-Following Data Augmentation

Instruction-data augmentation methods synthesize new instruction-response pairs rather than selecting from an existing pool. Self-instruct (Wang et al., 2023) and WizardLM (Xu et al., 2024a) bootstrap diverse instructions from a seed set, while AutoIF (Dong et al., 2025) and UltraIF (An et al., 2025) synthesize verifiable constraints from prompts and rejection-sample for compliance. RGDC inverts this synthesize-then-filter pattern, extracting constraints from existing reference translations so each is feasible by construction; Phase 2E filtering is then residual generation hygiene, not feasibility enforcement. A detailed comparison with AutoIF and UltraIF is in Appendix A, Table 8.

2.3 Machine Translation with LLMs

Recent work (Xu et al., 2024b,c; Alves et al., 2024; Bawden and Yvon, 2023) shows that general-purpose LLMs can match dedicated neural MT systems. For Thai specifically, Typhoon (Pipatanakul et al., 2023, 2024), SeaLLMs (Nguyen et al., 2024; Zhang et al., 2025), and Hunyuan-MT (Zheng et al., 2025) are representative recent efforts. Standard MT metrics like BLEU (Papineni et al., 2002), chrF (Popović, 2015), and COMET (Rei et al., 2020) score translation quality but not rule compliance. Our work targets Thai-English IF-MT, where prior systems either support IF without specialized translation quality or deliver translation quality without modeling prompt-level constraints.

2.4 LLM-as-Judge Evaluation

LLM-as-judge protocols correlate well with human preferences (Zheng et al., 2023) but exhibit length bias (Saito et al., 2023) and position bias (Wang et al., 2024). AlpacaEval v2 (Dubois et al., 2024) addresses length bias via a generalized linear model fitted on each evaluation. We adopt this protocol with an open-weight 35B LLM judge for reproducibility, cross-validated by two further open-weight judges from independent model families (Section 5.3) and by native-Thai human raters.

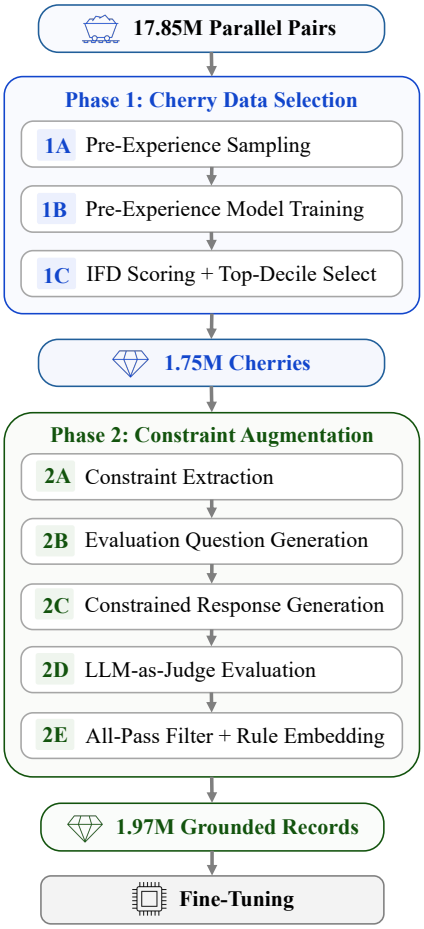


Figure 1: The two-phase RGDC pipeline. Phase 1 uses IFD scoring to cherry-select 1.75M instances from the 17.85M-record English-Thai parallel pool. Phase 2 extracts reference-grounded constraints and keeps only all-pass candidates, yielding the 1.97M-record Grounded dataset that trains the ChindaMT family.

3 Methodology

3.1 Pipeline Overview

Reference-Grounded Data Curation (RGDC) builds training data in two phases, shown in Figure 1. Phase 1 scores a large English-Thai parallel pool and keeps the hardest-but-learnable records. Phase 2 extracts verifiable constraints from each kept reference translation, so every constraint is feasible by construction, then regenerates the output under those constraints and keeps only generations that satisfy all of them. This yields the Grounded dataset, on which we fine-tune open-weight bases to produce the ChindaMT family.

3.2 Phase 1: Cherry Data Selection

Our base model is Qwen3.5-4B (Qwen Team, 2025). Following Li et al. (2024b), we briefly fine-tune it on a representative subset, then score every pool record by IFD and keep the top decile.

Pre-experience subset (1A–1B). For each corpus-direction pair, the base model computes mean-pooled sentence embeddings and per-instance perplexity. We cluster the samples and draw a fixed quota from each cluster within a mid-perplexity band, avoiding trivially easy and noisy extremes. The base model is then briefly freeze-tuned on this subset, exposing it to the domain without overfitting individual instances.

IFD scoring and selection (1C). The pre-experienced model scores every record by IFD, its output-loss ratio with and without the instruction,

$$\mathrm{IFD} = \frac{\ell(y \mid I, x)}{\ell(y)},$$

where ℓ is mean token-level cross-entropy, y the target, I the instruction, and x the source. An IFD below 1 means the instruction eases prediction; at or above 1 it does not help, so we discard them. From the rest we keep the **top decile by IFD**, the hardest-but-learnable band, yielding ∼1.75M cherry samples (per-sub-step counts in Table 2).

3.3 Phase 2: Constraint Augmentation

As in AutoIF (Dong et al., 2025) and UltraIF (An et al., 2025), we use a larger auxiliary LLM for constraint extraction, regeneration, and judging. From the 1.75M cherry samples, we extract constraints from each reference target (2A), generate yes/no evaluation questions (2B), regenerate translations under those constraints (2C), judge each constraint (2D), and retain all-pass candidates (2E).

Constraint extraction (2A). For each cherry sample, the auxiliary LLM receives the source, the target translation, and a taxonomy of five constraint categories. It returns constraints the target already demonstrates in Table 1, guaranteeing feasibility by construction. Full prompt in Appendix B.1.

Category	Example
Content	"include the term 'clean energy'"
Numerical	"use exactly one sentence"
Stylistic	"use formal Thai without slang"
Format	"return only the translation, no quotes"
Linguistic	"avoid English contractions"

Table 1: Example constraints for the five categories used by the Phase 2A extractor.

Evaluation question generation (2B). All constraints from a sample are batched into a single call to the auxiliary LLM that returns one yes/no question per constraint. Full prompt in Appendix B.2.

Constrained response generation (2C). Each cherry sample yields two candidate variants. The *sampled* variant embeds 1–3 randomly selected constraints, and the *all* variant embeds every extracted constraint. Each variant appends its constraints as a `Rules:` block to the source prompt, and the auxiliary LLM generates a new translation. System prompt in Appendix B.3.

LLM-as-judge evaluation (2D). For each candidate variant, the auxiliary LLM answers the per-constraint yes/no questions from 2B, marking each satisfied or violated. These judgments drive the all-pass filter in 2E. Full prompt in Appendix B.4.

All-pass filtering (2E). Only candidate variants where *every* constraint receives YES are retained, yielding the **Grounded dataset** (∼1.97M records; exact counts in Table 2). The retained constraints are embedded into the training instruction as a `Rules:` block, producing clean (instruction, input, output) triples for supervised fine-tuning. A reference-free quality audit finds these regenerated targets at least as good as their source-corpus references (Appendix D).

4 Experimental Setup

4.1 Source Parallel Corpora

Ten English-Thai parallel corpora (Appendix C, Table 9) supply both RGDC phases. After unification into Alpaca (`instruction, input, output`) format, the pool contains ∼17.85M records across both directions. Each source appears in both en-th and th-en for IFD scoring. Evaluation items drawn from this pool are held out of both phases from the outset (Section 4.3).

4.2 Grounded Training Dataset

The RGDC pipeline produces a single dataset we call **Grounded**, 1.97M records pairing each source and its `Rules:` block with the constraint-compliant translation. A worked record appears in Appendix D, with per-sub-step counts in Table 2. Retained records carry 3.7 constraints on average, and 56.5% of candidates survive to Grounded.

4.3 Evaluation Datasets

We construct two evaluation suites that share the Plain and Constrained formats but differ in data distribution, each with 400 samples per format, 200 en-th and 200 th-en. **CoreEval**, the primary suite, covers five deployment-relevant domains, government, finance, tech, UI, and miscellaneous, sampled from the parallel pool and length-stratified following Pipatanakul et al. (2024). **BroadEval** is independently curated for generalization, spanning ten broader domains such as news, literary, and social text, with about 25% paragraph-length inputs.

Sub-step	Records
Phase 1: Cherry selection	
1C. Cherry-selected (top-decile)	1,751,962
Phase 2: Constraint augmentation	
2A. Constraint extractions	1,751,962
2B. Evaluation questions	1,747,959
2C. Augmented prompts (2 variants)	3,495,704
2E. All-pass filtered (**Grounded**)	**1,973,358**

Table 2: Record counts at each RGDC sub-step, starting from a 17.85M parallel pool. Sub-steps 1B and 2D do not change record counts.

Both formats use English prompt scaffolding regardless of direction, matching the training format. Plain contains only the translation instruction and source, while Constrained adds a `Rules:` block of 1–4 constraints from the five categories in Table 1. Full templates in Appendix E.

4.4 Models Compared

Our four ChindaMT variants share one fine-tuning recipe (Section 4.5) on Grounded, differing only in base. Three use Qwen3.5 at 4B, 2B, and 0.8B for the deployment scaling axis, plus Qwen3-4B for cross-generation transfer. We compare each against size-matched Thai-capable MT specialists, Typhoon-Translate-1.5 (Pipatanakul et al., 2024), Hunyuan-MT-1.5 (Zheng et al., 2025), GemmaX2-28 (Cui et al., 2025), TranslateGemma (Finkelstein et al., 2026), and MiLMMT-46 (Shang et al., 2026), and against each variant's own base. Per-tier pairings are in Appendix F, Table 11.

4.5 Implementation Details

Phase 1. Sampling (1A) uses K-means with 100 clusters over base mean-pooled embeddings, drawing 10 per cluster within the 25^{th}–75^{th} perplexity band (40,683 samples). Training (1B) freeze-tunes for 1 epoch with the top 4 transformer blocks unfrozen (FP32, AdamW lr 2×10^{-5}, inverse-square-root schedule, 1% warmup, effective batch 64).

Phase 2. The four LLM-using sub-steps share a single auxiliary LLM, Qwen3.5-35B-A3B-FP8 with thinking disabled, using the role-specific decoding parameters in Appendix G, Table 12.

Fine-tuning. Full-parameter SFT for one epoch on Grounded, no preference optimization or RL. Core

hyperparameters in Appendix G, Table 13. The same recipe trains all four variants, with only the batch-size decomposition differing per tier (4×8 for 4B and Qwen3-4B, 8×4 for 2B, 16×2 for 0.8B, each on two GPUs, preserving effective batch 64). Runs in LLaMA-Factory (Zheng et al., 2024) on H100 under DeepSpeed ZeRO-2 with token packing and NEFTune noise $\alpha = 2$.

4.6 Evaluation Protocol

Judge. We use a single judge, Qwen3.6-35B-A3B-FP8 with greedy decoding and thinking disabled, to score every pair in the AlpacaEval-v2 format under three criteria in priority order, accuracy and faithfulness, then instruction and format compliance, then fluency. Full prompt in Appendix H.

Metric. From these preferences we report length-controlled win rate (LC%) (Dubois et al., 2024), where the AlpacaEval-v2 GLM corrects for length bias (Saito et al., 2023; Wang et al., 2024), with standard errors defined in Appendix I. Raw win rates appear in Table 15, external quality metrics in Section 5.2.

Judge-family bias. The primary judge shares the Qwen family with ChindaMT and Typhoon-Translate, and LLM judges can mildly favor their own lineage (Zheng et al., 2023), so Section 5.3 re-judges every comparison with two open-weight judges from independent families.

Split semantics. Plain has only the translation instruction and source, so its win rate reflects translation quality and output format, with no explicit rule to follow (Table 3). Constrained adds a rules block, so its win rate additionally reflects explicit rule compliance, under the shared template only, since no baseline exposes a `Rules:` slot. Translation quality on its own is measured directly by the external metrics in Table 4.

5 Results

5.1 Main Pairwise Results

CoreEval. ChindaMT-4B outperforms every same-tier and larger baseline on Mean, on both plain and constrained translation, and whether baselines use the shared scaffold or their default templates. Against same-size Typhoon-Translate-4B it reaches LC% 61.8 on Plain (shared) and 68.4 on Constrained, and Plain (own) LC% stays above 50 for every baseline, ruling out a prompt-format artifact. Constrained win rates over the larger HY-MT-1.5-7B approach the ceiling. ChindaMT-2B and ChindaMT-0.8B outperform or match every size-matched baseline, HY-MT-1.5-1.8B, GemmaX2-28-2B, and MiLMMT-46-1B. Per-direction breakdowns are in Appendix J.

BroadEval. The CoreEval pattern holds out of domain. ChindaMT-4B and ChindaMT-2B lead every reportable size-matched comparison without a loss, ChindaMT-4B ahead of same-size Typhoon-Translate-4B with LC% of 53.1 to 63.2 and of the larger HY-MT-1.5-7B by wider margins, while ChindaMT-0.8B leads its size-matched baselines except for a tie with GemmaX2-28-2B within 0.6 of 50. The instruction-following gap is the most durable out of domain, with Constrained LC over the larger HY-MT-1.5-7B reaching 98.2.

5.2 External-Benchmark Validation

We test whether the LC gains cost raw translation quality on FLORES-200 devtest (Wikipedia) and WMT24++ en-th (news), reporting four reference-free and reference-based metrics that correlate with human judgment, CometKiwi (Rei et al., 2022), GEMBA-DA (Kocmi and Federmann, 2023b), GEMBA-MQM (Kocmi and Federmann, 2023a), and MetricX-24 (Juraska et al., 2024). GEMBA-DA and GEMBA-MQM reuse our Qwen3.6-35B judge, with definitions and prompts in Appendices I and K. Each model runs under its best-case prompt, ChindaMT with the Grounded format and each baseline with its own model-card template.

ChindaMT-4B ties the top of FLORES Larger Mean (90.7, with MiLMMT-46-4B) and outperforms same-size Typhoon-Translate-4B on most comparisons across both benchmarks, while ChindaMT-2B leads its tier on FLORES (88.9) and ties for the WMT24++ lead (85.5). On WMT24++ Larger, ChindaMT-4B (87.8) trails HY-MT-1.5-7B (89.4) and sits level with MiLMMT-46-4B (87.9), the 1.6-point gap falling on the news-domain th-en direction where the larger 7B specialist holds an edge. Pairwise judging (Table 3) shows ChindaMT-2B leads size-matched comparisons, so external-metric strength does not cost instruction following. Full numbers in Appendix J, Table 15.

5.3 Cross-Judge and Human Validation

We check that LC rankings are not a primary-judge artifact, via cross-judge re-evaluation across independent families and blinded native-Thai rating.

	Model	Comparator	Plain (shared)			Plain (own)			Constrained		
			en→th	th→en	Mean	en→th	th→en	Mean	en→th	th→en	Mean
CoreEval	**ChindaMT-4B**	MiLMMT-46-4B	88.8	83.7	**86.3**	62.8	42.1	**53.3**	89.9	88.7	**89.5**
		TranslateGemma-4B	88.7	75.1	**81.0**	80.1	54.5	**66.7**	86.4	91.0	**87.2**
		Typhoon-Translate-4B	64.9	58.5	**61.8**	62.1	53.4	**57.8**	67.4	71.0	**68.4**
		HY-MT-1.5-7B	68.2	83.0	**74.0**	68.2	59.3	**62.9**	96.6	99.5	**97.9**
		GemmaX2-28-9B	94.7	92.8	**93.6**	66.2	45.0	**56.6**	94.4	93.5	**94.1**
	ChindaMT-2B	MiLMMT-46-1B	88.5	88.7	**88.4**	62.9	47.5	**54.6**	93.0	94.3	**93.7**
		HY-MT-1.5-1.8B	73.9	79.5	**75.2**	78.3	70.9	**72.6**	82.7	80.7	**78.6**
		GemmaX2-28-2B	92.9	87.3	**90.2**	70.4	38.9	**55.6**	87.5	88.6	**89.9**
	ChindaMT-0.8B	MiLMMT-46-1B	85.9	81.3	**83.4**	55.6	47.9	**50.9**	93.5	87.3	**90.3**
		HY-MT-1.5-1.8B	62.9	69.7	**64.6**	67.7	63.5	**63.4**	74.1	68.8	**67.8**
		GemmaX2-28-2B	92.2	82.3	**87.5**	62.0	39.4	**51.2**	89.2	82.1	**86.5**
BroadEval	**ChindaMT-4B**	MiLMMT-46-4B	—	—	—	49.0	62.1	**56.2**	—	—	—
		TranslateGemma-4B	80.4	83.7	**81.8**	72.5	62.5	**67.0**	74.6	80.3	**76.1**
		Typhoon-Translate-4B	60.0	56.6	**58.7**	53.3	49.8	**53.1**	63.4	63.1	**63.2**
		HY-MT-1.5-7B	51.3	87.8	**70.1**	57.5	68.5	**62.0**	96.7	99.7	**98.2**
		GemmaX2-28-9B	—	—	—	66.8	66.0	**67.4**	—	—	—
	ChindaMT-2B	MiLMMT-46-1B	91.8	97.4	**94.7**	70.3	66.5	**68.7**	—	—	—
		HY-MT-1.5-1.8B	53.4	90.1	**70.8**	72.6	83.0	**76.2**	73.5	88.3	**78.6**
		GemmaX2-28-2B	—	—	—	67.3	63.9	**66.3**	—	—	—
	ChindaMT-0.8B	MiLMMT-46-1B	89.0	95.9	**92.6**	46.1	53.9	**50.2**	—	—	—
		HY-MT-1.5-1.8B	34.8	76.1	**54.4**	47.8	67.7	**56.5**	57.7	74.0	**61.9**
		GemmaX2-28-2B	—	—	—	44.8	51.2	49.4	—	—	—

Table 3: Main pairwise LC% on CoreEval and BroadEval. (shared) uses a uniform prompt scaffold, (own) lets each baseline use its own template. **Bold** marks Means above 50. Shared-prompt comparisons where the comparator returns almost no target-language output (GemmaX2 BroadEval, MM-46-4B BroadEval, and Constrained BroadEval for MM-46-1B at the 2B and 0.8B tiers) are omitted.

Cross-judge. We re-judge every comparison in Table 3 with two additional open-weight judges from distinct families, GPT-OSS-120B (OpenAI, 2025) and Llama-3.3-70B-Instruct (Grattafiori et al., 2024), neither sharing lineage with any baseline. Constrained agreement is decisive, with ChindaMT WR reaching the mid-90s against the strongest MT specialists on both suites and judge spread under three points. Plain is judge-sensitive on MM-46 and GemmaX2 (WR% in $[45, 74]$), consistent with their parity on external metrics (Table 4). All 16 Constrained and 18 Plain (shared) comparisons agree on direction, and the three direction-disagreeing comparisons of 56 are near-parity Plain (own) within five points of 50%.

Human validation. Three native Thai raters scored blinded Plain and Constrained pairs from the ChindaMT-4B against Typhoon-Translate-1.5-4B comparison, Plain with the baseline template and Constrained with the shared template, full protocol in Appendix M. ChindaMT-4B is preferred on both splits, with stronger intensity on Constrained ($+0.68$) than Plain ($+0.39$). Inter-rater κ averages 0.73 (substantial) and judge-vs-human-majority $\kappa = 0.41$ (moderate), supporting the LLM judge as a population-level proxy. Agreement is stronger on Constrained (0.62) than on Plain (0.18), where the own-template condition leaves the two systems close (Table 3) and little signal for either judge or rater to separate.

All four channels, the primary LC% (Table 3), external metrics (Table 4), three-judge cross-judge (Table 5), and human evaluation (Table 6), agree on direction, so the headline does not rest on any single validation channel.

5.4 Method Lift and Parameter Scaling

We measure two scaling axes, own-base lift and within-Qwen3.5 parameter scaling at fixed recipe (Appendix N, Table 16). All four Qwen fine-tunes outperform their bases on CoreEval, with Plain/Constrained LC% of $55.7/62.2$ (Qwen3.5-4B), $71.9/69.5$ (Qwen3-4B), $77.1/74.2$ (Qwen3.5-2B), and $81.5/83.5$ (Qwen3.5-0.8B). Lift grows as parameters decrease, and Qwen3-4B gains more than Qwen3.5-4B from the same supervision, reflecting more headroom in smaller and older bases (Zhou et al., 2023a; Chen et al., 2024).

Within-family scaling is asymmetric. Plain lifts compress as size gaps narrow ($+12.6$ for 4B vs 2B,

		Model	CK↑		DA↑		MQM↑		Quality Mean↑	MtX↓		MtX Mean↓
			en→th	th→en	en→th	th→en	en→th	th→en		en→th	th→en	
FLORES-200	Larger 4B–9B	MiLMMT-46-4B	83.88	**84.98**	87.94	91.90	97.01	98.23	**90.7**	2.14	**1.97**	2.06
		TranslateGemma-4B	82.76	84.01	82.44	87.62	95.24	97.52	88.3	2.45	2.11	2.28
		Typhoon-Translate-4B	83.62	83.97	87.52	87.79	97.23	97.74	89.6	2.14	2.26	2.20
		HY-MT-1.5-7B	**84.38**	84.61	88.71	89.48	**97.63**	97.97	90.5	**1.98**	1.98	**1.98**
		GemmaX2-28-9B	82.43	84.45	85.20	**91.91**	96.94	98.34	89.9	2.45	2.07	2.26
		ChindaMT-4B (ours)	83.80	84.55	**89.32**	90.33	97.53	**98.39**	**90.7**	2.13	1.98	2.06
	Smaller 0.8B–2B	MiLMMT-46-1B	82.56	**84.41**	75.93	84.79	94.06	96.69	86.4	2.57	2.28	2.43
		HY-MT-1.5-1.8B	83.03	83.65	79.39	82.64	95.95	96.41	86.8	**2.22**	2.25	**2.24**
		GemmaX2-28-2B	81.80	84.25	79.78	**88.06**	95.54	**97.75**	87.9	2.66	**2.23**	2.45
		ChindaMT-0.8B (ours)	82.51	83.41	78.71	79.54	95.27	95.66	85.9	2.50	2.55	2.53
		ChindaMT-2B (ours)	**83.40**	84.08	**85.76**	86.24	**96.86**	97.25	**88.9**	2.25	**2.23**	**2.24**
WMT24++	Larger 4B–9B	MiLMMT-46-4B	80.30	**82.15**	84.39	89.21	94.74	96.68	87.9	3.14	3.08	3.11
		TranslateGemma-4B	78.55	80.51	77.21	84.31	93.14	95.92	84.9	3.37	3.28	3.33
		Typhoon-Translate-4B	79.46	80.62	81.82	85.21	95.39	97.40	86.6	3.29	3.44	3.37
		HY-MT-1.5-7B	**81.15**	81.90	**87.19**	**91.21**	**96.67**	**98.08**	**89.4**	**2.62**	**2.80**	**2.71**
		GemmaX2-28-9B	77.64	81.10	75.72	89.21	94.39	97.65	86.0	3.95	3.32	3.64
		ChindaMT-4B (ours)	80.23	81.21	83.85	87.60	96.63	97.36	87.8	3.21	3.19	3.20
	Smaller 0.8B–2B	MiLMMT-46-1B	77.84	**81.17**	69.44	82.80	92.18	95.36	83.1	3.69	3.42	3.56
		HY-MT-1.5-1.8B	**79.58**	80.93	77.44	83.94	94.86	96.16	**85.5**	**2.93**	**3.08**	**3.01**
		GemmaX2-28-2B	77.06	80.81	70.77	**84.99**	93.14	**96.38**	83.9	4.07	3.53	3.80
		ChindaMT-0.8B (ours)	77.80	79.58	69.65	74.28	92.05	94.14	81.2	3.74	3.71	3.73
		ChindaMT-2B (ours)	79.57	80.58	**79.48**	82.25	**95.02**	96.12	**85.5**	3.30	3.39	3.35

Table 4: External-metric translation quality on FLORES-200 (Wikipedia) and WMT24++ en-th (news), with CometKiwi (CK), GEMBA-DA (DA), and GEMBA-MQM (MQM) as 0–100 quality scores (↑); **Mean**↑ is their unweighted six-value average. MetricX-24 (MtX, ↓) is the native error score on a 0–25 scale (lower better), kept separate with its own **Mean**↓ over the two directions. Rows ordered by size within each tier, ChindaMT last. **Bold** marks the best per column within a tier, underline the second, ties bolded jointly.

+6.9 for 2B vs 0.8B), while Constrained lifts stay roughly constant per halving (+14.2 and +14.4). IF capability therefore scales sub-linearly and is largely preserved at 2B, making ChindaMT-2B a useful intermediate deployment choice.

Cross-family transfer. Fine-tuned on Grounded with only the base and chat template changed, the Gemma-3-4B fine-tune (FT) outperforms the Gemma-3-4B base with Plain/Constrained LC% of 99.1/66.6 (Table 16). All three judges agree on direction. The Plain margin is near ceiling because the base embeds its translation in English commentary that the rubric penalizes. The Constrained margin additionally reflects rule compliance. Since Phase 1 selection used the Qwen3.5-4B scorer, Grounded transfers across model families without base-specific selection.

5.5 Component Ablations

Each Qwen3.5-4B ablation drops one RGDC component from the full recipe and shares all other hyperparameters with ChindaMT-4B (Table 7). *No-cherry* replaces IFD selection with uniform random sampling from the 17.85M pool, *no-all-pass* drops the Phase 2E filter, and *no-rules* strips the `Rules:` block from training input.

Cherry selection contributes 8 LC on Plain only. The full recipe lifts +8.24 over no-cherry on Plain (outside SE) and +0.69 on Constrained (within SE). External metrics on out-of-domain FLORES and WMT24++ stay around ±0.3 on dataset Means (Appendix O), so Phase 1 aligns outputs with the CoreEval distribution rather than raising translation quality uniformly. Constraint compliance comes from Phase 2 supervision.

All-pass filter contributes a small positive lift. The full recipe gains +1.09/ + 1.06 over no-all-pass, modest relative to SE but consistent in sign on both axes, with external-metric deltas small on both benchmarks (Appendix O). The filter is data hygiene that complements reference grounding.

Rule-block training contributes 10–13 LC. The full recipe lifts +12.63/ + 9.53 over no-rules, above SE on every comparison, and no-rules

Model	Comparator	Plain	Constr.
ChindaMT-4B	MM-46-4B	51.0	89.6
	TG-4B	63.3	85.1
	Typhoon-4B	56.1	65.5
	HY-MT-7B	57.2	95.3
	GemmaX2-9B	56.4	93.7
ChindaMT-2B	MM-46-1B	54.1	91.3
	HY-MT-1.8B	62.1	73.4
	GemmaX2-2B	56.0	91.9
ChindaMT-0.8B	MM-46-1B	51.7	88.2
	HY-MT-1.8B	57.0	65.1
	GemmaX2-2B	53.0	90.1

Table 5: Cross-judge ChindaMT WR% on CoreEval, averaged across the three open-weight judges. Plain uses each comparator's own template, and values above 50 favor ChindaMT. Comparators are abbreviated, with full names in Table 11. BroadEval, Plain (shared), and per-judge numbers in Appendix L.

Metric	Plain	Constrained	Overall
ChindaMT win rate	0.64	0.69	0.67
Pref. intensity	+0.39	+0.68	+0.53
Inter-rater κ	0.74	0.70	0.73
Human-vs-judge κ	0.18	0.62	0.41

Table 6: Human validation by three native Thai raters on 50 Plain and 50 Constrained items from the ChindaMT-4B against Typhoon-Translate-1.5-4B comparison, Plain with the baseline template and Constrained with the shared template. Win rate averages ChindaMT preference across raters (ties as half-wins), and intensity is the mean -2 to 2 rating. Inter-rater κ averages three pairs, and human-vs-judge κ compares against the human majority.

loses all six quality-metric comparisons with Mean down 0.7–0.9 (Appendix O, Table 18). No-rules hurts Plain at least as much as Constrained, so rule-conditioned supervision aligns the broader output distribution with high-quality reference style rather than a narrow IF mechanism.

6 Analysis

6.1 Translation–IF Trade-off

Translation-specialized models forfeit instruction-following as they gain translation quality (Xu et al., 2024b; Alves et al., 2024), and the pattern recurs across our baselines. Tables 4 and 5 show MM-46-4B, TranslateGemma-4B, and GemmaX2 at parity with ChindaMT on plain translation, yet they lose every Constrained comparison under all three judges. Typhoon-Translate-4B occupies the opposite corner, with weaker plain translation and a smaller Constrained gap.

	Components			Full recipe lift	
Variant	Cherry	Filter	Rules	Plain	Constr.
Full	✓	✓	✓	—	—
no-cherry	×	✓	✓	+8.24	+0.69
no-all-pass	✓	×	✓	+1.09	+1.06
no-rules	✓	✓	×	+12.63	+9.53

Table 7: RGDC component ablations at 4B on CoreEval. Full recipe lift is the LC% margin over each ablation in pairwise judging, above the 50 tie. SE $[2.1, 2.5]$.

RGDC avoids both failure modes because rule and translation supervision share one source of truth, the reference translations that cherry selection retains. The Phase 2 ablation (Section 5.5) supports the mechanism, since removing the Rules block reduces Plain LC more than Constrained LC, pointing to broad output-distribution alignment rather than a narrow instruction-following circuit. Plain-translation parity with the strongest size-matched MT specialists is the central evidence that the instruction-following gains do not cost translation quality.

On external metrics ChindaMT reaches parity rather than dominance, the one residual specialist edge being in-domain news, where the larger 7B HY-MT model leads on WMT24++ th-en (Table 4). The deployment-relevant gains, rule compliance and cross-domain robustness at a fixed size, are where ChindaMT leads.

6.2 Qualitative Case Analysis

Consider a Thai-to-English translation under the rule *"Return only the translated text"*. ChindaMT-4B and its Qwen3.5-4B base both follow the rule; Typhoon-Translate echoes the rule block and an `EN:` prefix before its translation, a direct rule violation. BLEU, chrF, and neural QE metrics would partially accept the Typhoon output because the trailing translation is correct, while LC% catches the format violation through the hard penalty in Appendix H. Full transcript in Appendix P.

This case instantiates the two failure modes that motivate the paper. Typhoon-Translate, a translation specialist, renders the sentence accurately but leaks the rule block and an `EN:` prefix, quality without compliance. The instruction-tuned Qwen3.5-4B base obeys the rule yet drops the source word *web*, compliance with a faithfulness loss. ChindaMT-4B alone does both, returning only the translation while preserving the content its base omits, resolving the trade-off in Section 6.1.

7 Conclusion

We introduced **ChindaMT**, an open-weight instruction-following Thai-English translation family at 4B, 2B, and 0.8B parameters, and **RGDC**, the two-phase pipeline that builds its training data. RGDC extracts every rule from a reference translation that already satisfies it, so feasibility is guaranteed by construction. Under length-controlled pairwise evaluation, ChindaMT outperforms or matches size-matched translation specialists at every tier on CoreEval and BroadEval, without sacrificing raw translation quality on FLORES-200 or WMT24++, a result that two open-weight cross-judges and native Thai raters corroborate. The recipe transfers across four bases and two Qwen generations. We release model weights, the Grounded dataset, evaluation suites, and code under open licenses.

8 Limitations

Reference-bounded constraint coverage. RGDC extracts only rules a reference already satisfies, so the constraints it can supervise are bounded by the reference distribution. Demands that no reference exemplifies, or that resist a yes/no check, fall outside the pipeline, trading breadth for feasibility. Term, length, register, and output-format instructions are covered. A mandated glossary and placeholder tokens fall outside by design, since reference grounding supervises only what a reference demonstrates. Widening coverage by grounding in the resources of a domain, or with synthetic constraint generation, is left to future work.

Single primary judge. Primary LC and GEMBA rely on one open-weight judge (Qwen3.6-35B), so absolute LC values may reflect its calibration. We mitigate this with external metrics (Table 4), AlpacaEval-v2 length and position controls, a three-family cross-judge (Table 5), and human raters (Table 6). Direction agreement holds across all three judge families on every Constrained and Plain (shared) comparison.

Language-pair and domain coverage. All experiments cover English-Thai in both directions only. Extension to other low-resource pairs, especially Southeast Asian languages, and to broader registers is left to future work.

Prompt-format asymmetry. The Grounded training format matches the Typhoon-Translate template but differs from HY-MT and GemmaX2. On Plain we evaluate baselines under both shared and own templates. Constrained uses the shared format only, since no baseline exposes a `Rules:` slot, omitting some GemmaX2 comparisons (Table 3).

Base-architecture coverage. All four ChindaMT bases are Qwen. The recipe transfers across Qwen3.5 and Qwen3 generations, and the curated data lifts Gemma-3-4B over its base (Table 16), though with Qwen-derived Phase 1 selection. Rerunning selection with the target base and transfer to other architectures remain untested.

9 Ethical Considerations

Data. RGDC uses ten public English-Thai corpora (Appendix C). Because several carry ShareAlike terms, the released Grounded dataset and evaluation suites adopt CC-BY-SA 4.0. The suites ship source text and model outputs but no reference translations or personally identifying information, consistent with the source licenses, and one BroadEval item retains the CC-BY-NC-4.0 terms of its source, marked in the released file.

Human evaluation. The study (Appendix M) used three native Thai speakers who volunteered without compensation and gave informed consent after being told the purpose and their right to withdraw. They rated anonymized pairs in a minimal-risk task recording only preferences, with no personal data collected. The study required no formal ethics-board review at our institution.

Intended use and risks. Like any MT system, ChindaMT can produce fluent but incorrect output or reflect training-data bias, so its outputs warrant human review in high-stakes use.

Release and compute. Model weights and code are released under Apache-2.0, the weights inheriting the Qwen base-model license, and the Grounded dataset and evaluation suites under CC-BY-SA 4.0, for reproducibility. The small 0.8B–4B models keep compute modest and accessible to the low-resource Thai-English community.

Acknowledgements

We thank the anonymous reviewers and the area chair for their constructive comments, SiamAI for computational resources, OpenThai Lab for its support, and the three native Thai speakers who volunteered as raters. An AI assistant was used for language editing, drafting support, and help with LaTeX and code. The authors reviewed and verified all content and take full responsibility for it.

References

Duarte M. Alves, José Pombal, Nuno M. Guerreiro, Pedro H. Martins, João Alves, Amin Farajian, Ben Peters, Ricardo Rei, Patrick Fernandes, Sweta Agrawal, Pierre Colombo, José G. C. de Souza, and André F. T. Martins. 2024. Tower: An open multilingual large language model for translation-related tasks. In *Conference on Language Modeling (COLM)*.

Kaikai An, Li Sheng, Ganqu Cui, Shuzheng Si, Ning Ding, Yu Cheng, and Baobao Chang. 2025. UltraIF: Advancing instruction following from the wild. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 18711–18726, Suzhou, China. Association for Computational Linguistics.

Rachel Bawden and François Yvon. 2023. Investigating the translation performance of a large multilingual language model: the case of BLOOM. In *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*, pages 157–170, Tampere, Finland. European Association for Machine Translation.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, and Hongxia Jin. 2024. AlpaGasus: Training a better alpaca with fewer data. In *International Conference on Learning Representations (ICLR)*, pages 34767–34797.

Christos Christodoulopoulos and Mark Steedman. 2015. A massively parallel corpus: the Bible in 100 languages. *Language Resources and Evaluation*, 49(2):375–395.

Menglong Cui, Pengzhi Gao, Wei Liu, Jian Luan, and Bin Wang. 2025. Multilingual machine translation with open large language models at practical scale: An empirical study. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5420–5443, Albuquerque, New Mexico. Association for Computational Linguistics.

Ona de Gibert, Graeme Nail, Nikolay Arefyev, Marta Bañón, Jelmer van der Linde, Shaoxiong Ji, Jaume Zaragoza-Bernabeu, Mikko Aulamo, Gema Ramírez-Sánchez, Andrey Kutuzov, Sampo Pyysalo, Stephan Oepen, and Jörg Tiedemann. 2024. A new massive multilingual dataset for high-performance language technologies. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 1116–1128, Torino, Italia. ELRA and ICCL.

Guanting Dong, Keming Lu, Chengpeng Li, Tingyu Xia, Bowen Yu, Chang Zhou, and Jingren Zhou. 2025. Self-play with execution feedback: Improving instruction-following capabilities of large language models. In *International Conference on Learning Representations (ICLR)*.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B. Hashimoto. 2024. Length-controlled AlpacaEval: A simple way to debias automatic evaluators. In *Conference on Language Modeling (COLM)*.

Ahmed El-Kishky, Adithya Renduchintala, James Cross, Francisco Guzmán, and Philipp Koehn. 2021. XLEnt: Mining a large cross-lingual entity dataset with lexical-semantic-phonetic word alignment. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10424–10430, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mara Finkelstein, Isaac Caswell, Tobias Domhan, Jan-Thorsten Peter, Juraj Juraska, Parker Riley, Daniel Deutsch, Geza Kovacs, Cole Dilanni, Colin Cherry, Eleftheria Briakou, Elizabeth Nielsen, Jiaming Luo, Kat Black, Ryan Mullins, Sweta Agrawal, Wenda Xu, Erin Kats, Stephane Jaskiewicz, and 2 others. 2026. TranslateGemma technical report. *arXiv preprint arXiv:2601.09012*.

Aaron Grattafiori et al. 2024. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Yuxin Jiang, Yufei Wang, Xingshan Zeng, Wanjun Zhong, Liangyou Li, Fei Mi, Lifeng Shang, Xin Jiang, Qun Liu, and Wei Wang. 2024. FollowBench: A multi-level fine-grained constraints following benchmark for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4667–4688, Bangkok, Thailand. Association for Computational Linguistics.

Juraj Juraska, Daniel Deutsch, Mara Finkelstein, and Markus Freitag. 2024. MetricX-24: The Google submission to the WMT 2024 metrics shared task. In *Proceedings of the Ninth Conference on Machine Translation*, pages 492–504, Miami, Florida, USA. Association for Computational Linguistics.

Tom Kocmi and Christian Federmann. 2023a. GEMBA-MQM: Detecting translation quality error spans with GPT-4. In *Proceedings of the Eighth Conference on Machine Translation*, pages 768–775, Singapore. Association for Computational Linguistics.

Tom Kocmi and Christian Federmann. 2023b. Large language models are state-of-the-art evaluators of translation quality. In *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*, pages 193–203, Tampere, Finland. European Association for Machine Translation.

Philipp Koehn. 2024. Neural methods for aligning large-scale parallel corpora from the web for south and East Asian languages. In *Proceedings of the Ninth Conference on Machine Translation*, pages 1454–1466, Miami, Florida, USA. Association for Computational Linguistics.

kvush. 2024. English-Thai texts. https://huggingface.co/datasets/kvush/english_thai_texts. Hugging Face dataset repository.

Ming Li, Yong Zhang, Shwai He, Zhitao Li, Hongyu Zhao, Jianzong Wang, Ning Cheng, and Tianyi Zhou. 2024a. Superfiltering: Weak-to-strong data filtering for fast instruction-tuning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14255–14273, Bangkok, Thailand. Association for Computational Linguistics.

Ming Li, Yong Zhang, Zhitao Li, Jiuhai Chen, Lichang Chen, Ning Cheng, Jianzong Wang, Tianyi Zhou, and Jing Xiao. 2024b. From quantity to quality: Boosting LLM performance with self-guided data selection for instruction tuning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7602–7635, Mexico City, Mexico. Association for Computational Linguistics.

Xian Li, Ping Yu, Chunting Zhou, Timo Schick, Omer Levy, Luke Zettlemoyer, Jason Weston, and Mike Lewis. 2024c. Self-alignment with instruction backtranslation. In *International Conference on Learning Representations (ICLR)*.

Pierre Lison and Jörg Tiedemann. 2016. OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 923–929, Portorož, Slovenia. European Language Resources Association (ELRA).

Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. 2024. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. In *International Conference on Learning Representations (ICLR)*.

Lalita Lowphansirikul, Charin Polpanumas, Attapol T. Rutherford, and Sarana Nutanong. 2022. A large English-Thai parallel corpus from the web and machine-generated text. *Language Resources and Evaluation*, 56(2):477–499.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. 2025. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *IEEE Transactions on Audio, Speech and Language Processing*, 33:3776–3786.

Xuan-Phi Nguyen, Wenxuan Zhang, Xin Li, Mahani Aljunied, Zhiqiang Hu, Chenhui Shen, Yew Ken Chia, Xingxuan Li, Jianyu Wang, Qingyu Tan, Liying Cheng, Guanzheng Chen, Yue Deng, Sen Yang, Chaoqun Liu, Hang Zhang, and Lidong Bing. 2024. SeaLLMs - large language models for Southeast Asia. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 294–304.

OpenAI. 2025. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Kunat Pipatanakul, Phatrasek Jirabovonvisut, Potsawee Manakul, Sittipong Sripaisarnmongkol, Ruangsak Patomwong, Pathomporn Chokchainant, and Kasima Tharnpipitchai. 2023. Typhoon: Thai large language models. *arXiv preprint arXiv:2312.13951*.

Kunat Pipatanakul, Potsawee Manakul, Natapong Nitarach, Warit Sirichotedumrong, Surapon Nonesung, Teetouch Jaknamon, Parinthapat Pengpun, Pittawat Taveekitworachai, Adisai Na-Thalang, Sittipong Sripaisarnmongkol, Krisanapong Jirayoot, and Kasima Tharnpipitchai. 2024. Typhoon 2: A family of open text and multimodal Thai large language models. *arXiv preprint arXiv:2412.13702*.

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.

Qwen Team. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

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.

Ricardo Rei, Marcos Treviso, Nuno M. Guerreiro, Chrysoula Zerva, Ana C Farinha, Christine Maroti, José G. C. de Souza, Taisiya Glushkova, Duarte Alves, Luisa Coheur, Alon Lavie, and André F. T. Martins. 2022. CometKiwi: IST-unbabel 2022 submission for the quality estimation shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 634–645, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Keita Saito, Akifumi Wachi, Koki Wataoka, and Youhei Akimoto. 2023. Verbosity bias in preference labeling by large language models. In *NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following*.

Yuzhe Shang, Pengzhi Gao, Wei Liu, Jian Luan, and Jinsong Su. 2026. Scaling model and data for multilingual machine translation with open large language models. *arXiv preprint arXiv:2602.11961*.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA).

Jörg Tiedemann. 2020. The Tatoeba translation challenge – realistic data sets for low resource and multilingual MT. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1174–1182, Online. Association for Computational Linguistics.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui. 2024. Large language models are not fair evaluators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9440–9450, Bangkok, Thailand. Association for Computational Linguistics.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. 2024a. WizardLM: Empowering large pre-trained language models to follow complex instructions. In *International Conference on Learning Representations (ICLR)*.

Haoran Xu, Young Jin Kim, Amr Sharaf, and Hany Hassan Awadalla. 2024b. A paradigm shift in machine translation: Boosting translation performance of large language models. In *International Conference on Learning Representations (ICLR)*.

Haoran Xu, Amr Sharaf, Yunmo Chen, Weiting Tan, Lingfeng Shen, Benjamin Van Durme, Kenton Murray, and Young Jin Kim. 2024c. Contrastive preference optimization: Pushing the boundaries of LLM performance in machine translation. In *International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pages 55204–55224. PMLR.

Wenxuan Zhang, Hou Pong Chan, Yiran Zhao, Mahani Aljunied, Jianyu Wang, Chaoqun Liu, Yue Deng, Zhiqiang Hu, Weiwen Xu, Yew Ken Chia, Xin Li, and Lidong Bing. 2025. SeaLLMs 3: Open foundation and chat multilingual large language models for Southeast Asian languages. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (System Demonstrations)*, pages 96–105.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623. Curran Associates, Inc.

Mao Zheng, Zheng Li, Bingxin Qu, Mingyang Song, Yang Du, Mingrui Sun, and Di Wang. 2025. Hunyuan-MT technical report. *arXiv preprint arXiv:2509.05209*.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, and Zheyan Luo. 2024. LlamaFactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 400–410, Bangkok, Thailand. Association for Computational Linguistics.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023a. LIMA: Less is more for alignment. In *Advances in Neural Information Processing Systems*, volume 36, pages 55006–55021. Curran Associates, Inc.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023b. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*.

A Detailed Comparison with Prior IF-Data-Augmentation Methods

Expanding Section 2.2, Table 8 compares RGDC against the two closest IF-data-augmentation methods, AutoIF (Dong et al., 2025) and UltraIF (An et al., 2025), across seven axes.

The axes isolate where RGDC departs from synthesize-then-filter methods. Extracting constraints from a reference that already satisfies them makes every target feasible by construction, whereas AutoIF and UltraIF synthesize from prompts and discard the unsatisfiable share. This admits an all-pass filter stricter than thresholding or rejection sampling, and grounding in a large parallel pool scales RGDC to 1.97M translation-specialized records, an order of magnitude beyond their general-domain corpora.

B RGDC Pipeline Prompts

All four prompts run on vLLM with `enable_thinking=false` and per-role sampling from Table 12, with auxiliary LLM `Qwen3.5-35B-A3B-FP8` (Section 4.5). The four

prompts chain into one Phase 2 record, with 2A extracting constraints, 2B turning each into a verification question, 2C regenerating under them, and 2D scoring compliance.

B.1 Constraint Extraction (2A)

Extracts verifiable constraints across five categories from a Simplified Query and its Answer (the existing reference translation). The five categories match the evaluation-time taxonomy (Table 1), and requiring each constraint to be verifiable and absent from the query keeps it a checkable signal rather than a restatement of the task. The full template follows, with its five in-context examples.

```
You are an expert in generating instruction constraints for a
      given simplified query.

Definition of Constraint: The smallest unit of restriction or
      requirement that can be added to an instruction to make
      the task more specific or challenging.

Your Task: Given a Simplified Query and its Answer, generate
      NEW constraints that could be added to the query. The
      generated constraints should be relevant to the query and
       the answer, and should make the task more specific
      without changing its fundamental goal.

Simplified Query: {QUERY}
Answer: {ANSWER}

Follow these steps:
1. Understand the basic goal of the simplified query.
2. Analyze the answer to understand what aspects can be
      constrained.
3. Generate NEW constraints that are:
   - Relevant to the query and answer
   - Verifiable in the response
   - Not already explicitly stated in the query
4. Categorize constraints into the following types:

Constraint Types:
- Content Constraints:
    - Specific Terms or Symbols: Mandatory use of certain terms
          or symbols (e.g., must include the word 'beautiful').
    - Required Elements or Concepts: Specific elements or
         concepts that must be included (e.g., must mention the
         Great Wall).
    - Thematic Directives: Instructions about thematic content,
         perspective, or focus (e.g., focus on environmental
         impact).

- Numerical Constraints:
    - Quantities related to the content: number of points,
         sentences, paragraphs, word count, or examples (e.g.,
         exactly three sentences, at least 5 examples).

- Stylistic Constraints:
    - Tone and style requirements (e.g., formal, informal,
         conversational, humorous).
    - Specific language or terminology preferences (e.g.,
         encyclopedic style, avoid jargon).

- Format Constraints:
    - Structure or format requirements (e.g., list, JSON,
         bullet points, table).
    - Programming language specifications (e.g., Java, Python).
    - Presentation styles (e.g., markdown, code block format).

- Linguistic Constraints:
    - Language specifications (e.g., in English, in Spanish).
    - Sentence structure requirements (e.g., use only simple
         sentences, imperative form).
    - Word-level requirements (e.g., lowercase, single-rhyme,
         alliteration).

Response Format:
- Generate constraints that are appropriate for the given query.
- Ensure generated constraints do not overlap with what is
      already in the query.
- Only include constraint types that are relevant.
- Present each constraint as a dictionary with a 'constraint'
      key.

```json
{
    "Content Constraints": [
        {"constraint": "..."},
        {"constraint": "..."}
    ],
    "Numerical Constraints": [...],
    "Stylistic Constraints": [...],
    "Format Constraints": [...],
    "Linguistic Constraints": [...]
}
```

Examples:

Example 1:
Query: "How do I check if a user pressed the cancel button on a
       prompt in JS?"
Answer: "You can check if a user pressed cancel by comparing
      the prompt result to null. When the user clicks cancel,
      the prompt() function returns null instead of a string.
      So you simply use: if (result === null) { /* user
      cancelled */ }"

Response:
```json
{
    "Numerical Constraints": [
        {"constraint": "Answer with exactly three sentences."}
    ]
}
```
Explanation: The query asks for a JavaScript explanation. A
      numerical constraint "exactly three sentences" is
      generated to add structure to the response. This
      constraint is verifiable by counting the sentences in the
       answer (which has exactly 3 sentences).

Example 2:
Query: "Explain quantum computing"
Answer: "Quantum computing is like having a super-powered
      calculator that can try many answers at once instead of
      one at a time. Regular computers use bits (0s and 1s),
      but quantum computers use 'qubits' that can be both 0 and
       1 at the same time. This lets them solve certain complex
       problems much faster than regular computers."

Response:
```json
{
    "Content Constraints": [
        {"constraint": "Explain in simple terms using analogies
              ."}
    ]
}
```
Explanation: The query asks for an explanation of quantum
      computing. A content constraint "explain in simple terms
      using analogies" is generated because the answer uses
      accessible language and analogies (like "super-powered
      calculator"). This makes the explanation requirement more
       specific without changing the core goal.

Example 3:
Query: "I want to replace string like 'This is a new {object}
      at {place}' with a Map, {object: 'student', place: 'point
       3, 4'}, and get a result 'This is a new student at point
       3, 4'. How can I do this?"
Answer: "public String replaceWithMap(String template, Map<
      String, String> values) {\n String result = template;\n
      for (Map.Entry<String, String> entry : values.entrySet())
       {\n result = result.replace('{{' + entry.getKey() + '}}',
       entry.getValue());\n }\n return result;\n}"

Response:
```json
{
```
```

```
    "Format Constraints": [
        {"constraint": "Answer using Java programming language
            ."}
    ]
}
```
Explanation: The query asks how to perform string replacement.
      The answer is provided in Java code. A format constraint
      "Answer using Java" is generated to specify the
      programming language requirement, making the query more
      precise about the expected output format.

Example 4:
Query: "Write a program which accepts a command line parameter
      as a question and sends it to a server via HTTP GET
      method."
Answer: "import sys\nimport requests\n\nif len(sys.argv) < 2:\n
       print('Usage: python script.py <question>')\n sys.exit
      (1)\n\nquestion = sys.argv[1]\nresponse = requests.get('
      http://example.com/api', params={'question': question})\
      nprint(response.text)"

Response:
```json
{
    "Format Constraints": [
        {"constraint": "Write as a Python program."}
    ]
}
```
Explanation: The query asks for a program without specifying
      the language. Since the answer is in Python, a format
      constraint "Write as a Python program" is generated to
      make the language requirement explicit. This ensures the
      response will be in Python.

Example 5:
Query: "Recommend books"
Answer: "- 'Dream of the Red Chamber' by Cao Xueqin - A classic
       tale of noble families\n- 'Journey to the West' by Wu
      Cheng'en - Adventures of a monk and his companions\n- '
      Romance of the Three Kingdoms' by Luo Guanzhong - Epic
      historical warfare\n- 'Water Margin' by Shi Nai'an -
      Tales of outlaws and heroes\n- 'The Scholar' by Wu Jingzi
       - Satire of the imperial examination system\n- 'Strange
      Tales from a Chinese Studio' by Pu Songling - Collection
      of supernatural stories\n- 'Border Town' by Shen Congwen -
       A love story set in rural China\n- 'Family' by Ba Jin -
      Story of generational conflict\n- 'Rickshaw Boy' by Lao
      She - Tale of a rickshaw puller's struggles\n- 'To Live'
      by Yu Hua - A man's journey through tumultuous times"

Response:
```json
{
    "Stylistic Constraints": [
        {"constraint": "Use Shakespeare's tone."}
    ],
    "Format Constraints": [
        {"constraint": "Use bullet points in the answer."}
    ],
    "Numerical Constraints": [
        {"constraint": "Recommend exactly ten books."}
    ],
    "Content Constraints": [
        {"constraint": "Focus on Chinese literature."}
    ]
}
```
Explanation: The query simply asks to "recommend books" which
      is very broad. Multiple constraints are generated based
      on the answer: (1) Stylistic constraint for "Shakespeare'
      s tone" adds a creative challenge; (2) Format constraint
      for "bullet points" matches the answer's structure; (3)
      Numerical constraint for "ten books" matches the count in
       the answer; (4) Content constraint for "Chinese
      literature" narrows the scope to match the answer's focus.

Please only provide the response in JSON format.
```

### B.2 Evaluation Question Generation (2B)

Converts each constraint from Appendix B.1 into a yes/no verification question, batched into one LLM call per record. Single yes/no questions make compliance objectively checkable in Phase 2D, and the empty-string fallback drops descriptive or embedded constraints so the filter acts only on genuine requirements.

```
You are an expert in crafting questions to evaluate whether a
      response to a query adheres to specific constraints.

For each constraint below, design a question that human
      evaluators can use to assess if the response meets that
      constraint. Each question should focus solely on its
      given constraint.

If a constraint is meaningless or is part of the content itself
       (e.g., descriptions, scenarios, examples), respond with
      an empty string for that constraint.

Example:
Query: Recommend books.
Constraints:
1. Use bullet points in your answer.
2. Recommend exactly ten books.

Response:
{
    "1": "Does the response use bullet points?",
    "2": "Does the response recommend exactly ten books?"
}

Now generate evaluation questions for:
Query: {query}

Constraints:
{constraints_list}

Respond only in JSON format mapping constraint numbers to
      questions:
{
    "1": "question for constraint 1",
    "2": "question for constraint 2"
}
```

B.3 Constrained Response Generation (2C)

Each cherry record produces two variants, *sampled* (1–3 constraints) and *all* (every extracted constraint). Each is formatted with a `Rules:` block appended to the original instruction and sent to the generator with the system prompt below. The two variants expose the model to varying rule counts, and the system prompt suppresses preamble so each target stays a clean translation.

System prompt.

```
You are an expert tasked with answering the given query. Please
       provide a clear and concise response directly, without
      introductory phrases such as 'What a great question,' '
      Here is the answer,' or similar expressions. Focus solely
       on addressing the query while strictly following its
      inside constraints.
```

User message template.

```
[Original Instruction]

Rules:
1. [Constraint 1]
2. [Constraint 2]

[Original Input]
```
```

Aspect	AutoIF	UltraIF	RGDC (ours)
Source data	36 hand-written seeds	ShareGPT prompts	Cherry-selected translation Q&A
Constraint origin	Self-instruct from seeds	Decomposed from prompts	Extracted from references
Verification	Python execution	LLM-as-judge	LLM-as-judge with explanations
Filter strictness	Score threshold	Rejection sampling	Every constraint must pass
Scale	10–25k	175k	1.97M
Aux. model	No	Yes (UltraComposer)	Yes (IFD scorer + aux LLM)
Domain	General	General	Translation

Table 8: Comparison of RGDC with AutoIF and UltraIF across seven axes.

B.4 Compliance Evaluation (2D)

The judge evaluates every yes/no question from Appendix B.2 against each candidate from Appendix B.3, returning `YES`/`NO` with explanation. Records pass the Phase 2E filter only if every question receives `YES`. Requiring a justification before each verdict reduces label noise, which matters because a single `NO` drops the record under the all-pass filter.

```
You are an expert that is good at judging whether the response
      to a given query meets the specified evaluator questions.
Your task is to carefully examine the response to determine if
      it adheres to each requirement outlined in the evaluator
      questions.

[Query] {query}
[Response] {response}
[Evaluator Question] {question}

For each question, please provide a justification for your
      evaluation, explaining how the response does or does not
      satisfy the criteria and a score ('YES' or 'NO')
      indicating whether the answer satisfies each constraint.

You should only respond in the following JSON format:
{
   "Question 1": {
      "explanation": "",
      "score": "YES" or "NO"
   },
   "Question 2": {
      "explanation": "",
      "score": "YES" or "NO"
   }
}
```

C Source Parallel Corpora

Table 9 lists the 10 publicly available English-Thai parallel corpora forming the ∼17.85M-record source pool, after unification into Alpaca (`instruction`, `input`, `output`) format. Phase 1 (Section 3.2) cherry-selects from this pool, and Phase 2 augments the selected subset.

Corpus	Type
`bible`	Religious parallel (Christodoulopoulos and Steedman, 2015)
`paracrawl`	Web crawl (Koehn, 2024)
`elrc`	Government / news (Tiedemann, 2012)
`hplt`	Web crawl (de Gibert et al., 2024)
`en-th-texts`	Mixed (kvush, 2024)
`opensubtitles`	Film/TV subtitles (Lison and Tiedemann, 2016)
`scb_2020`	Mixed (Lowphansirikul et al., 2022)
`tatoeba`	Crowdsourced pairs (Tiedemann, 2020)
`wikimedia`	Encyclopedic (Tiedemann, 2012)
`xlent`	Web-mined (El-Kishky et al., 2021)

Table 9: The 10 publicly available English-Thai parallel corpora (all en↔th) unified into the RGDC training pool, used by Phase 1 (cherry selection) and Phase 2 (constraint augmentation).

D Grounded Dataset Record Example and Quality Audit

Record example. A Grounded record in Alpaca (`instruction`, `input`, `output`) format. `instruction` combines task header, source, and `Rules:` block; `input` is empty; `output` is the constraint-compliant translation.

```
[instruction]
Task: Translate Thai -> English.
Source:
จะส่งไปในอีก 35 นาทีค่ะ
### Response:

Rules:
- Use a customer service professional style.
- The translation must consist of exactly one sentence.
- Include the number 35 in the output.

[output]
The item will be dispatched in 35 minutes.
```

`[instruction]` and `[output]` mark the Alpaca fields and are not stored.

Quality of the regenerated targets. We score 20,000 regenerated targets, 5,000 per direction and variant, and their source-corpus references with CometKiwi (Rei et al., 2022), which is independent of our judge. The regenerated targets match or exceed their references on 85.9% of records overall and on more than 80% in each direction (Table 10). Since reference-free estimation can mildly favor

fluent output, we read this as no quality regression from constraint conditioning. The retained constraints are also mostly semantic, with content, linguistic, and stylistic categories at 77% against format and numerical at 23%.

Slice	n	ΔCK	SE	hyp ≥ ref
Overall	20,000	+13.10	0.09	85.9%
en→th	10,000	+16.03	0.13	90.9%
th→en	10,000	+10.18	0.13	80.8%

Table 10: CometKiwi audit of Phase 2C regeneration. ΔCK is the mean paired difference between the regenerated target and its source-corpus reference, and hyp ≥ ref is the share of records where the regenerated target scores at least as high.

E Evaluation Prompt Templates

Plain and Constrained prompt formats from Section 4.3. The scaffolding is identical across directions, and for the th-en direction, `Translate English to Thai.` and `EN:` become `Translate Thai to English.` and `TH:`.

Plain.

```
Translate English to Thai.

EN: <source text>
```

Constrained.

```
Translate English to Thai.
Rules:
- Return only the translated text
- Use a clear, professional tone in Thai
- Keep all numerals in Arabic digits

EN: <source text>
```

F Models Compared

Table 11 lists the ChindaMT variants we evaluate and the external baselines used in the main results.

Model	Size
ChindaMT (ours)	
ChindaMT (Qwen3.5, 4B primary)	4B, 2B, 0.8B
ChindaMT-Qwen3-4B (cross-gen)	4B
External baselines	
Typhoon-Translate-1.5	4B
Hunyuan-MT-1.5	7B, 1.8B
GemmaX2-28	9B, 2B
TranslateGemma	4B
MiLMMT-46	4B, 1B

Table 11: Models compared. TranslateGemma-4B and MiLMMT-46-4B are 4B-only comparators.

G Implementation Hyperparameters

Three hyperparameter sets cover the pipeline: Phase 2 auxiliary-LLM decoding (Table 12), shared fine-tuning across all four variants (Table 13), and test-time inference (below).

Compute. All fine-tuning ran on two NVIDIA H100s under DeepSpeed ZeRO-2, about 68 wall-clock hours across the four variants (roughly 25 hours per 4B model, 10 for 2B, and 8 for 0.8B), and the cross-family Gemma-3-4B run about 7 hours. Phase 1 took about two and a half days on one H100, mostly IFD scoring of the pool. Phase 2 data generation, the dominant cost, ran on the Qwen3.5-35B-A3B-FP8 vLLM endpoint over roughly two weeks of active generation.

Sub-step	Temp.	Max tokens
2A. Constraint extraction	0.7	1024
2B. Evaluation questions	0	512
2C. Constrained responses	0.7	512
2D. Judge evaluation	0	1024

Table 12: Temperature and max-token settings used by the auxiliary LLM at each Phase 2 sub-step. Identical across runs and bases.

Parameter	Value
Method	SFT (full parameters)
Epochs	1
Learning rate	2×10^{-5}
Scheduler	inverse-square-root, 1% warmup
Weight decay	0.01
Optimizer	AdamW (β_1=0.9, β_2=0.999)
Effective batch size	64 (2 GPUs)
Cutoff length	1024
Seed	42

Table 13: Core ChindaMT-4B fine-tuning hyperparameters. The same recipe is used for all four variants.

Test-time inference. Our fine-tunes and their bases decode with temperature 0.01, top-p 0.7, top-k 20, repetition penalty 1.05, and up to 1024 new tokens, run as batched HuggingFace inference on a single H100. External baselines use per-model settings, temperature 0.2 and top-p 0.9 for GemmaX2 and Typhoon-Translate, 0.7 and 0.6 for HY-MT-1.5, greedy decoding for MiLMMT-46, and up to 512 new tokens for TranslateGemma. The same configurations are used for the LC% pairwise tables (Section 4.6) and the external-metric benchmarks (Section 5.2).

H Evaluation Judge Prompt

Prompt for the pairwise LC judge (Qwen3.6-35B-A3B-FP8) used in every pairwise table of Section 5. Output follows the AlpacaEval-v2 `ranking_parser` format with a three-tier rubric and a hard penalty for any content beyond the translation. Decoded with temperature 0, `enable_thinking=false`, max_tokens 1024.

```
<|im_start|>system
You are an expert evaluator for translation quality. Evaluate
      based on:

A) Accuracy & Faithfulness (highest priority)
- Translate correctly, accurately, and completely according to
      the given content.
- Preserve ALL factual information, meaning, numbers, names,
      and intent.
- Do NOT add, omit, distort, or hallucinate any content.

B) Instruction Compliance & Format Preservation (equally
      critical)
- STRICTLY follow every instruction and rule in the original
      prompt.
- Preserve formatting/structure exactly when required (line
      breaks, punctuation, tags, placeholders, bulleting,
      spacing, casing, etc.).
- If the instruction requires specific constraints (e.g., "
      output ONLY the translation" / "no extra words" / "keep
      proper nouns"), enforce them.

C) Naturalness & Fluency (only after A and B)
- Translation should read naturally in the target language,
      like a human translation.
- Correct grammar, appropriate register, and smooth phrasing.

HARD PENALTY RULE (very important):
- Judge ONLY the translated text in each model output.
- If a model output contains ANY extra content that is not part
       of the translation (e.g., explanations, apologies, notes,
       metadata, quotes of the prompt, commentary, headings, "
      Here is the translation:", etc.), it MUST receive a
      significant penalty.
- If the instruction says the output must contain ONLY the
      translation, then ANY extra words make that output worse,
       even if the translation itself is good.

TIE-BREAKERS:
1) Fewer instruction/format violations wins.
2) If still tied, the more accurate/complete one wins.
3) If still tied, the more natural/fluently phrased one wins.
<|im_end|>
<|im_start|>user
I will give you the instructions (prompts) given to the models,
       and the responses of two models. Please rank the models
      based on which responses would be preferred by humans and
       above criteria. All inputs and outputs should be python
      dictionaries.

Here is the prompt:
{
    "instruction": """{instruction}""",
}

Here are the outputs of the models:
[
    {
        "model": "model_1",
        "answer": """{output_1}"""
    },
    {
        "model": "model_2",
        "answer": """{output_2}"""
    }
]

Now please rank the models by the quality of their answers, so
      that the model with rank 1 has the best output based on
      the criteria. Then return a list of the model names and
      ranks, i.e., produce the following output:
[
    {'model': <model-name>, 'rank': <model-rank>},
    {'model': <model-name>, 'rank': <model-rank>}
]

Your response must be a valid Python list and should contain
      nothing else because we will directly execute it in
      Python.
Do NOT use markdown code blocks or ```. Return ONLY the raw
      Python list.
<|im_end|>
```

I Evaluation Metrics Reference

Definitions, strengths, and table pointers for every quality metric used in the paper.

I.1 Raw win rate (WR%)

Share of items where the judge prefers the target over the reference, with ties as half-wins. Given N items with wins w, losses ℓ, draws d:

$$\text{WR\%} = (w + 0.5 \cdot d) \,/\, N \cdot 100.$$

WR% summarizes per-item votes and shares units with LC%, but inherits length bias because LLM judges prefer verbose responses (Saito et al., 2023; Dubois et al., 2024). Used in Table 15, and inside every LC% comparison.

I.2 Length-controlled win rate (LC%)

The win rate the target would have at matched reference length. AlpacaEval-v2 fits a logistic GLM predicting `preference` from a standardized length difference $\Delta\ell$ and per-instruction difficulty d,

$$P(\text{target wins} \mid \text{item}) = \sigma(\alpha \cdot \tanh(\Delta\ell) + \beta \cdot d + \gamma),$$

and evaluates it at $\Delta\ell = 0$, giving $\text{LC\%} = 100 \cdot \mathbb{E}[P(\text{target wins} \mid \Delta\ell{=}0)]$. LC% removes the length bias, is stable at $N{=}400$, and agrees with WR% within 1–2 points. It needs $\sim$400 items per comparison, so Table 15 reports WR% per direction. The per-direction cells of Table 3 are separate fits on each direction, so the Mean is not their average. Standard errors are the AlpacaEval-v2 estimate over the raw per-item votes; cross-judge (Section 5.3) and external win-rate comparisons use a 1000-draw bootstrap. Used in Tables 3 and 16.

I.3 External translation-quality metrics

Table 4 reports four metrics, with the GEMBA prompts in Appendix K.

CometKiwi (Rei et al., 2022) is a neural reference-free quality estimator in $[0, 100]$ (`Unbabel/wmt22-cometkiwi-da`) that rewards neither reference-copying nor rule compliance.

GEMBA-DA (Kocmi and Federmann, 2023b) prompts our Qwen3.6-35B judge for a 0–100 score

from source, reference, and candidate, making it reference-dependent and prone to LLM biases.

GEMBA-MQM (Kocmi and Federmann, 2023a) uses the same judge to label MQM error spans by severity, aggregated to a 0–100 score, and is reference-free and diagnostic but coarse.

MetricX-24 (Juraska et al., 2024) is a neural reference-based MQM error score on a native 0–25 scale (lower is better); we report it in this native form and exclude it from the external-metric Mean.

I.4 Metric alignment with LC%

Our LC% judge (Appendix H) scores pairwise on *accuracy* > *instruction compliance* > *fluency* with a hard penalty for extra non-translation content. Table 14 compares LC% to the external metrics across five axes.

Metric	Acc.	IF	Flu.	Pair.	Ref.
WR% / LC%	yes	yes	yes	yes	no
CometKiwi	yes	no	yes	no	no
GEMBA-DA	yes	indirectly	yes	no	yes
GEMBA-MQM	yes	no	via Flu.	no	no

Table 14: Metric coverage across accuracy (Acc.), instruction-following (IF), fluency (Flu.), pairwise comparison (Pair.), and reference dependence (Ref.). LC% covers all five, our primary metric. External metrics (Section 5.2) check translation quality alone.

J Per-Direction Pairwise Win Rates

Table 15 reports per-direction win rates by data source. The CoreEval columns break the shared-prompt comparisons of Table 3 into directions; the external columns apply the same protocol (Section 4.6) to FLORES and WMT24++, which carry plain translation only. The external pattern matches Table 4: ChindaMT-4B and ChindaMT-2B lead all size-matched FLORES comparisons and most on WMT24++, with the HY-MT news-domain edge clearest against the larger 7B specialist on WMT24++ th-en.

K External-Benchmark Metric Prompts

CometKiwi is a neural model and uses no prompt. For GEMBA-DA and GEMBA-MQM we adapt the prompts of Kocmi and Federmann (2023b) and Kocmi and Federmann (2023a) to our Qwen3.6-35B judge. GEMBA-DA keeps the direct-assessment format and returns a 0–100 score from source, reference, and candidate. GEMBA-MQM is zero-shot with six coarse categories, and its labeled errors are aggregated into a 0–100 score by subtracting 1, 5, and 10 per minor, major, and critical error. On a parse failure the judge is re-asked with a one-line format reminder. The two prompts follow, with placeholders in braces.

GEMBA-DA prompt.

```
You are evaluating the quality of a machine translation.

Source ({src_lang}):
{src}

Reference translation ({tgt_lang}):
{ref}

Candidate translation ({tgt_lang}):
{mt}

Rate the candidate's overall translation quality on a scale
      from 0 to 100, where:
- 0-30: major meaning errors, poor fluency, unusable
- 30-60: partial meaning preserved, noticeable errors
- 60-80: good translation, minor issues
- 80-100: excellent translation, near-perfect

Consider accuracy, fluency, and faithfulness to the source. The
       reference is one acceptable translation; equally valid
      alternatives should not be penalized.

Output only a single integer between 0 and 100, nothing else.
```

GEMBA-MQM prompt.

```
You are performing a Multidimensional Quality Metrics (MQM)
      evaluation.

Source ({src_lang}):
{src}

Candidate translation ({tgt_lang}):
{mt}

Identify translation errors in the candidate. For each error,
      output one line in the exact format:
  CATEGORY / SEVERITY / short-description

Categories: Accuracy, Fluency, Terminology, Style, Locale,
      Other
Severities: Minor, Major, Critical

If there are no errors, output the single line:
  NO_ERRORS

Do not include any text other than the error lines.
```

L Per-Judge Cross-Judge Numbers

Table 17 reports the per-judge WR% underlying the consensus mean in Table 5. Columns Pri, OSS, and Llama are the three open-weight judges, Qwen3.6-35B (primary), GPT-OSS-120B, and Llama-3.3-70B-Instruct, with identical prompts.

M Human Evaluation Protocol

Three native Thai raters, one with a translation background, each rated 100 stratified items from the ChindaMT-4B against Typhoon-Translate-1.5-4B comparison, Plain with the baseline template and Constrained with the shared template, 50 Plain and 50 Constrained, A/B-blinded on a 5-point (+2 to −2) preference scale toward ChindaMT. The

Model	Comparator	CoreEval				External (plain)			
		Plain		Constrained		FLORES		WMT24++	
		en→th	th→en	en→th	th→en	en→th	th→en	en→th	th→en
ChindaMT-4B	MiLMMT-46-4B	88.8	84.5	90.5	90.0	55.3	52.5	51.4	39.9
	TranslateGemma-4B	86.8	72.5	83.8	87.5	73.9	64.7	71.9	53.3
	Typhoon-Translate-4B	65.0	59.3	66.0	69.8	55.4	60.9	55.1	49.5
	HY-MT-1.5-7B	65.5	78.5	94.5	98.5	61.0	60.8	48.3	38.9
	GemmaX2-28-9B	95.0	92.5	95.5	94.8	67.5	58.2	68.0	44.8
ChindaMT-2B	MiLMMT-46-1B	88.5	87.0	92.5	93.5	–	–	–	–
	HY-MT-1.5-1.8B	72.3	75.0	80.0	74.3	73.8	76.5	64.7	53.1
	GemmaX2-28-2B	93.5	89.3	93.5	91.3	67.9	54.7	67.8	47.1
ChindaMT-0.8B	MiLMMT-46-1B	86.0	79.5	93.0	86.0	–	–	–	–
	HY-MT-1.5-1.8B	62.5	64.5	72.0	62.0	57.0	59.7	45.1	38.0
	GemmaX2-28-2B	93.0	85.0	93.5	86.0	50.3	38.0	49.4	31.8

Table 15: Per-direction pairwise win rate (WR%) by data source. CoreEval columns break the shared-prompt comparisons of Table 3 into directions (Plain and Constrained); external columns apply the same pairwise protocol (Section 4.6) to FLORES and WMT24++, which carry plain translation only. Dashes mark comparisons not run, as MiLMMT-46-1B has no external pairing. CoreEval columns report raw WR% per direction; external bootstrap SE is $[1.3, 1.6]$.

win rate collapses this to win/tie/loss by sign (ties as half-wins) and intensity is the mean rating. Two attention-check items per rater were excluded. Results are in Table 6.

Rater instructions (translated from Thai). For each pair, A and B in random order (Constrained items also show a Rules block), raters chose the better translation, judging in priority order accuracy and faithfulness, then rule compliance, then fluency, and scored it 2 (clearly better), 1 (slightly better), or 0 (tie). A few items were attention checks with an obviously incorrect option. Participation was voluntary with the right to withdraw, and only preferences were recorded. After unblinding, preferences are oriented toward ChindaMT for the $+2$ to -2 scale in Table 6.

N Method Lift and Parameter Scaling

Referenced from Section 5.4. Table 16 reports two scaling axes on CoreEval, each RGDC fine-tune against its own base (upper block), one step of within-Qwen3.5 parameter halving at fixed recipe (middle block), and the cross-family Gemma-3-4B fine-tune (last row).

Model	Comparator	Plain	Constr.
ChindaMT-4B	Qwen3.5-4B	55.7	62.2
ChindaMT-Qwen3-4B	Qwen3-4B	71.9	69.5
ChindaMT-2B	Qwen3.5-2B	77.1	74.2
ChindaMT-0.8B	Qwen3.5-0.8B	81.5	83.5
ChindaMT-4B	ChindaMT-2B	62.6	64.2
ChindaMT-2B	ChindaMT-0.8B	56.9	64.4
Gemma-3-4B FT	Gemma-3-4B	99.1	66.6

Table 16: LC% on CoreEval. Upper block: each ChindaMT variant vs its own base; middle block: ChindaMT variants across parameter halving; last row: the cross-family Gemma-3-4B fine-tune (FT) vs its base. SE $[1.8, 2.4]$, $[2.2, 2.4]$, and $0.7/2.3$ by block, the last small because the Plain cell is near ceiling. Under the primary, GPT-OSS-120B, and Llama-3.3-70B judges, the Gemma-3-4B row has Plain/Constrained WR% of $97.9/67.4$, $96.2/62.4$, and $97.9/67.0$ (n=400).

O External-Metric Ablations at 4B

Table 18 reports direction-averaged external metrics per RGDC ablation, complementing the LC% ablations in Table 7.

Cherry selection. No-cherry slightly edges the full recipe out of domain, with Mean lifts of 0.23 on FLORES (from 90.65 to 90.88) and 0.31 on WMT24++ (from 87.81 to 88.12). The 8.24 Plain CoreEval gain (Section 5.5) therefore reflects in-domain alignment, not a uniform quality boost.

All-pass filter. The full recipe holds the higher Mean on both datasets (FLORES 90.65 vs 90.60, WMT24++ 87.81 vs 87.79), with per-metric deltas scattering near zero without a consistent direction.

Rule-block training. No-rules loses on every metric in both datasets, with Means dropping 0.92 on FLORES (from 90.65 to 89.73) and 0.72 on WMT24++ (from 87.81 to 87.09). The largest drops come from GEMBA-DA, indicating translation-quality regression rather than narrow instruction-following loss.

	Model	Comparator	Plain (shared)			Plain (own)			Constrained		
			Pri	OSS	Llama	Pri	OSS	Llama	Pri	OSS	Llama
CoreEval	**ChindaMT-4B**	MiLMMT-46-4B	86.6	86.5	86.4	55.1	47.1	50.9	90.2	89.5	89.0
		TranslateGemma-4B	79.6	77.8	78.6	66.5	59.9	63.5	85.6	83.2	86.4
		Typhoon-Translate-4B	62.1	55.0	54.1	59.8	52.2	56.2	67.9	66.1	62.6
		HY-MT-1.5-7B	72.0	61.6	65.5	62.1	53.0	56.6	96.5	94.8	94.8
		GemmaX2-28-9B	93.8	93.0	92.2	59.4	54.1	55.6	95.1	92.8	93.1
	ChindaMT-2B	MiLMMT-46-1B	87.8	85.8	84.8	55.6	51.9	54.9	93.0	90.8	90.2
		HY-MT-1.5-1.8B	73.6	60.0	66.1	70.4	54.6	61.4	77.1	70.9	72.1
		GemmaX2-28-2B	91.4	90.3	89.1	58.4	54.0	55.6	92.4	91.0	92.4
	ChindaMT-0.8B	MiLMMT-46-1B	82.8	84.5	84.5	52.6	51.2	51.1	89.5	87.5	87.5
		HY-MT-1.5-1.8B	63.5	58.5	61.0	62.3	51.0	57.8	67.0	62.5	65.8
		GemmaX2-28-2B	89.0	89.8	88.0	54.4	50.1	54.6	89.8	90.3	90.3
BroadEval	**ChindaMT-4B**	MiLMMT-46-4B	—	—	—	59.4	51.6	66.6	—	—	—
		TranslateGemma-4B	81.6	81.1	77.1	68.5	64.6	69.5	74.2	73.5	73.2
		Typhoon-Translate-4B	60.0	53.2	56.8	56.4	50.9	58.9	62.8	55.9	61.5
		HY-MT-1.5-7B	66.5	67.8	71.2	59.0	52.5	59.2	96.0	96.2	95.0
		GemmaX2-28-9B	—	—	—	73.2	61.2	73.5	—	—	—
	ChindaMT-2B	MiLMMT-46-1B	95.0	95.0	94.8	70.9	65.0	67.4	—	—	—
		HY-MT-1.5-1.8B	68.8	66.8	72.0	73.4	66.2	68.1	75.1	73.2	72.8
		GemmaX2-28-2B	—	—	—	69.5	60.6	73.2	—	—	—
	ChindaMT-0.8B	MiLMMT-46-1B	93.0	93.2	92.8	52.4	49.8	57.6	—	—	—
		HY-MT-1.5-1.8B	51.0	51.5	59.2	51.5	51.2	55.8	60.1	63.8	64.4
		GemmaX2-28-2B	—	—	—	52.4	45.0	59.9	—	—	—

Table 17: Per-judge WR% for the three judges behind the Table 5 consensus. Direction-averaged Mean; values above 50 favor ChindaMT. n=400, bootstrap SE $[0.9, 2.6]$.

Dataset	Variant	CK	DA	MQM	Mean
FLORES-200	Full recipe	84.18	89.83	97.96	90.65
	No-cherry	84.28	90.27	98.09	**90.88**
	No-all-pass	84.19	89.68	97.92	90.60
	No-rules	83.36	88.12	97.70	89.73
WMT24++	Full recipe	80.72	85.73	97.00	87.81
	No-cherry	80.89	86.57	96.89	**88.12**
	No-all-pass	80.76	85.90	96.70	87.79
	No-rules	80.27	84.43	96.56	87.09

Table 18: External-metric ablations at 4B, direction-averaged; ablations defined in Section 5.5. **Bold** marks the best Mean per dataset.

P Qualitative Case Transcript

Full input and per-model outputs for the rule-violation example in Section 6.2.

```
INSTRUCTION:
Translate Thai to English.
Rules:
- Return only the translated text
- Use concise, natural English

TH: ท่องเว็บได้แบบส่วนตัวในอุปกรณ์ทุกเครื่อง
ChindaMT-4B:
  Browse the web privately on any device.
Qwen3.5-4B:
  Browse privately on any device.
Typhoon-Translate:
  Rules:
  - Return only the translated text
  - Use concise, natural English

  EN: Browse the web privately on every device.
```