

SubVision@WMT26: QLoRA Fine-Tuning of Hy-MT2-1.8B for Chinese to English Video Subtitle Translation

Prajwal Ghimire

Department of Artificial Intelligence
Kathmandu University
prajwalghimire22@gmail.com

Aashish Mahato

Department of Artificial Intelligence
Kathmandu University
aashishmtho24@gmail.com

Abstract

We describe **SubVision**, our submission to the WMT26 Video Subtitle Translation shared task. The system fine-tunes the 1.8B-parameter multilingual translation model Hy-MT2-1.8B (Zheng et al., 2026) using 4-bit QLoRA (Dettmers et al., 2023) on 60,000 sentence pairs sampled from the TVsub Chinese-to-English subtitle corpus (Wang et al., 2018). The training pipeline uses a single instruction style prompt template, LoRA adapters (rank 16, $\alpha = 32$) on all attention and MLP projections, and early stopping based on dev set sacreBLEU. We fix the decoding with beam search with beam size 4, repetition penalty 1.15, and no-repeat 3-gram constraints. The fine-tuned model on a 200 pair TVsub test split with multiple references achieves sacreBLEU 37.63, chrF 52.76, and TER 53.93, compared with zero-shot NF4 baselines of sacreBLEU 11.60 for Hy-MT2-1.8B and 15.79 for Hy-MT2-7B on the same test pairs, prompt, and decoding configuration. The same configuration is used for official inference, translating subtitle lines independently while preserving SRT timing. We report our full training and inference configuration, describe a data processing detail in parsing the corpus’s multi-document, multi-reference SGM files, and compare our final configuration against nine additional trained configurations that differ in training pair count, sequence-length budget, LoRA rank, and decoding generation length. The paired bootstrap analysis over this comparison shows that given the 200 segment test set size, the nominal ranking of the submitted configuration is not statistically distinguishable from several of the alternatives. We report this analysis along with the point estimates rather than presenting the ranking as an established result. We note that evaluation uses an in-domain TVsub split. The code is available at: <https://github.com/praaajg/wmt26-hymt2>.

1 Introduction

The WMT26 Video Subtitle Translation shared task asks participants to translate the spoken dialogue of video subtitles, distributed as SRT files, from Chinese into English (WMT26 Organizers, 2026). Subtitle translation differs from sentence level document translation in that each line is short, colloquial, and time boxed, and the organizers additionally provide per-video metadata (descriptions, character lists, and glossaries) that participants may use to disambiguate character names and recurring terminology across a video.

Our system **SubVision** takes a compute constrained approach built around a single publicly released small model. We adapt Hy-MT2-1.8B (Zheng et al., 2026) which is a 1.8B-parameter multilingual translation model, by fine-tuning it with QLoRA (Dettmers et al., 2023) on the TVsub corpus (Wang et al., 2018) which is a large Chinese-to-English corpus of subtitles drawn from Chinese television dramas. The shared task itself supplies no training or validation data, only the unlabeled demonstration and official test files described in Section 3.5. Therefore, participants must source their own training corpus. All the training and evaluation of automatic metrics uses TVsub. We then apply the resulting fine-tuned model directly to the organizer provided official test files using the same translation prompt and decoding configuration used in training and evaluation, without incorporating the accompanying video metadata for the shared task submission itself.

We underscore this scope explicitly because it shapes how our results should be read. We report BLEU, chrF, and TER numbers that come from an in-domain TVsub test split with references, not from the official test set, for which we have no references and therefore no automatic score. Our contribution is consequently a fully specified, reproducible fine-tuning, decoding, and inference pipeline for a small, resource friendly subtitle translation model, together with an honest account of what our internal evaluation does and does not tell us about performance on the shared task’s actual test data. Our contributions are:

- A resource efficient QLoRA fine-tuning pipeline that adapts Hy-MT2-1.8B to

Chinese–English subtitle translation using 60,000 TVsub sentence pairs and a single 23.7 GB GPU.

- An empirical comparison showing a substantial improvement from task specific adaptation over zero-shot Hy-MT2-1.8B under an identical prompt, evaluation split, and decoding configuration, together with a comparison against nine additional trained configurations.
- A paired bootstrap analysis of the ten fine-tuned configurations, showing that the nominal differences among the strongest systems are not statistically distinguishable at the 200 segment test set size.
- A fully specified SRT inference pipeline for the official WMT26 test files, including preservation of subtitle indices and timestamps, together with the released LoRA adapter and inference resources.
- A documented correction to the multi-document SGM reference parsing procedure used for all reported evaluation, addressing segment-ID collisions across document boundaries.

2 Related Work

Subtitle machine translation. Subtitle translation introduces constraints on brevity, timing, and register that distinguish it from general document or sentence level machine translation (Matusov et al., 2019). Matusov et al. (2019) show that NMT systems adapted specifically for subtitling outperform general purpose systems by better matching the length and reading speed constraints of on screen text. The recent subtitling systems explicitly optimize for these constraints via length controlled MT that condenses translations to respect characters per line, lines per block and reading speed limits without large quality loss (Petrick et al., 2025). For movie subtitles, prompting LLMs with movie meta information such as title, summary and genre improves translation accuracy, with the movie title giving the most consistent gains (Pramodya et al., 2025). Video context can also improve subtitle translation when the textual input is ambiguous. Li et al. (2023) found that incorporating paired video clips improves translation over text only models, particularly for ambiguous subtitles (Li et al., 2023). The TVsub corpus (Wang et al., 2018), which we use for fine-tuning, provides large scale Chinese-to-English subtitle pairs drawn from television dramas and captures the colloquial, turn taking register typical of subtitle dialogue.

Document level context in MT. A natural extension for subtitle and dialogue translation is to condition on surrounding context. Maruf et al. (2019) survey document level neural MT methods, architectures, training and decoding strategies, and evaluation. Cross-sentence history has been shown to improve Chinese-English NMT by up to +2.1 BLEU points (Wang et al., 2017). MTCue shows that unstructured document and metadata context in dialogue improves translation quality (+0.88 BLEU, +1.58 COMET over a non-contextual baseline) and enables zero-shot control of attributes such as formality (Vincent et al., 2023). At the same time, document level evaluation finds that contextual information is pervasive across sentences but does not always change human judgments (Kim, 2025).

Parameter efficient fine-tuning. Domain adaptation is important because vanilla NMT performs poorly under domain mismatch when in-domain corpora are scarce (Chu and Wang, 2018). LoRA (Hu et al., 2021) adapts large pretrained models by injecting trainable low rank matrices into frozen weight projections, substantially reducing the memory required for fine-tuning. Adapter modules originate as compact alternatives attaining within 0.4% of full fine-tuning on GLUE while adding only 3.6% parameters per task (Houlsby et al., 2019), and for NMT specifically tiny task specific adapter layers match full fine-tuning for domain adaptation at a fraction of per-task cost (Bapna and Firat, 2019). QLoRA (Dettmers et al., 2023) extends LoRA by additionally quantizing the frozen base model to 4-bit NF4 precision with double quantization which is what makes fine-tuning a 1.8B-parameter model practical on a single GPU with limited memory in our setup. Two stage QLoRA curriculum fine-tuning has been shown to substantially improve low-resource MT, e.g. chrF++ from 22.73 to 41.10 over NLLB-200 on an unseen language (Umar et al., 2026). We use the PagedAdamW8bit optimizer, building on 8-bit block-wise quantized optimizers that maintain 32-bit performance at a fraction of the memory footprint (Dettmers et al., 2021), for the same reason.

Instruction style translation prompting. Instruction tuning (Ouyang et al., 2022) is now a standard way of aligning language models to task-specific behavior. Instruction finetuning has also been shown for NMT itself which allows a single small NMT model to follow multiple customization instructions (formality, multi domain, multi modal translation) at GPT-3.5-level quality (Rau-nak et al., 2024). We follow this convention by framing translation as a single turn chat completion with an explicit instruction, rather than a raw sequence-to-sequence mapping.

3 Dataset and Preprocessing

3.1 Corpus

The WMT26 Video Subtitle Translation shared task provides no training or validation data of its own: the only organizer released material is the unlabeled demonstration package and official test files (Section 3.5). Therefore, any training corpus is a participant’s own choice. We use the TVsub corpus (Wang et al., 2018) as this training proxy, cloned directly from the corpus’s public repository. The TVsub dataset provides pre-split, pre-processed Chinese–English subtitle pairs under `data/processed/{train,dev,test}`, with a plain-text source file for each split and a multi-reference SGM file for the dev and test references.

3.2 Sampling and Filtering

The training pairs from the processed training file we build as follows. Each source and target line is detokenized (Section 3.3) and discarded if either side is empty after cleaning, if either side exceeds 512 characters (twice our configured maximum subtoken length), or if the character length ratio between target and source falls outside $[0.3, 4.0]$, which removes badly misaligned pairs. The filtered pairs are shuffled with a fixed random seed (42) and truncated to 60,000 training pairs. The same detokenization (but not the ratio/length cutoffs, which apply only to training) is applied when building the 500-pair development split used for early stopping and best-checkpoint selection during training, both of which additionally require overlapping segment IDs between the source file and the parsed multi-reference file.

All test set metrics and qualitative examples reported in this paper (Table 7 onward) come from an evaluation script run against the saved best adapter, using its own default of 200 pairs from the same `data/processed/test/` split (the first 200 segment IDs with both a source line and at least one parsed reference). The same 200 source lines and multi-reference sets are reused without modification for the zero shot baseline evaluations of the un-adapted base models reported in Section 6.5.

3.3 Detokenization

TVsub’s released text contains subword-joiner artifacts (`@-@`, `@@`) and HTML escaped entities from its original crawl. Our detokenizer removes joiner artifacts, HTML-unescapes the text, collapses repeated whitespace, corrects spacing around punctuation, and applies Unicode NFKC normalization. This detokenization is applied uniformly to training pairs, dev/test sources, and references before scoring so that reported metrics are computed on cleaned text throughout.

3.4 Multi-Reference SGM Parsing

The dev and test reference files are provided as SGM markup with segments grouped under `<DOC>` blocks and numbered with `<seg id="...">` tags. Two `<DOC>` blocks (docids SUB01 and SUB02) each restart segment numbering at 1, since each block corresponds to a different source document (television episode) in both files. A parser that reads `<seg id="...">` without also tracking which `<DOC>` block it belongs to will silently conflate segments from the two episodes that share the same local ID. Our parser instead iterates over `<DOC>` blocks in file order, accumulates an offset equal to the maximum local segment ID seen so far, and adds that offset to each segment’s local ID to obtain a corpus-level ID that is unique across the whole file. This is applied identically when building the dev and test reference sets. We flag it here because an earlier, pre-fix version of our own pipeline lacked this offsetting. It corrupted the dev-set sacreBLEU that drives early stopping and best-checkpoint selection during training, as well as the final test report, motivating the corrected parser used throughout this paper. The nine locally trained configurations catalogued in Section 6.3 and SubVision, were designed and run only after the fix and never used the flawed offsetting. The same corrected parser is used to build the reference set for the zero shot baseline evaluations in Section 6.5, so the fine-tuned and baseline numbers we compare are scored against an identical, correctly offset reference set.

3.5 Official Test Set

The shared task additionally distributes a demonstration package for the official test set, consisting of a per-video `*_zh.srt` subtitle file and a meta-data record. Neither the demonstration package nor the full official test set includes reference translations: both are unlabeled source side releases, consistent with a shared task that scores submissions after the fact rather than allowing participants to self score against the official test data. For the example video `b4101o4ev1s` (episode 2 of the drama 万福金安 *Wànfú Jīn’ān*, lit. “ten-thousandfold blessings, golden peace”), the meta-data record provides the video and album titles and an album level synopsis. The source file contains 231 subtitle blocks. Table 9 (Appendix A) shows several consecutive source lines exactly as provided by the organizers, with Pinyin and an English gloss added for readers who do not read Chinese, illustrating the short, dialogue style utterances and period register (forms of address, court vocabulary) typical of this genre.

4 Methodology

4.1 Base Model and Quantization

We fine-tune `tencent/Hy-MT2-1.8B` (Zheng et al., 2026), quantized to 4-bit NF4 precision with double quantization via `bitsandbytes`, using a `bfloat16` compute dtype where supported. The LoRA adapters (Hu et al., 2021) of rank $r = 16$ and scaling factor $\alpha = 32$ with dropout 0.05 are attached to all attention and MLP projection matrices (query, key, value, output, gate, up, and down projections). The quantized base weights remain frozen throughout training and only the adapter weights are trainable. Section 6.5 evaluates this same 4-bit NF4 quantization of the base model, without any adapter attached, as a zero shot reference point.

4.2 Model Choice and Task Constraints

We adapt `tencent/Hy-MT2-1.8B` rather than a larger general-purpose LLM for reasons tied directly to the shared task’s rules (WMT26 Organizers, 2026). The Video Subtitle Translation task runs a constrained track only: any submitted model, or the sum of an ensemble’s weights, must total under 20B parameters, and teams must release the fine-tuned weights under a license permitting at least unrestricted non-commercial use. The organizers explicitly recommend the Hy-MT2 family as a strong base model for fine-tuning within this constraint, and teams that use a Hunyuan-MT-derived base model and achieve competitive results are additionally eligible for the task’s monetary prizes. We chose the 1.8B-parameter variant specifically within that family because it lets the entire pipeline, quantization, QLoRA training, early stopping, and inference, run on a single 23.7 GB GPU, which matches our available hardware far more comfortably than the 7B variant would while still leaving substantial parameter budget headroom under the task’s 20B cap. It also keeps the artifact we must open-source small: because only the LoRA adapter (rank 16) needs to be redistributed on top of the frozen, publicly available base model. Our released weights are ~74 MB (Open Weights, below) rather than a multi-gigabyte merged checkpoint. This choice is not merely a compute budget default: the reduced budget model configuration sweep in Section 6.4 fine-tuned Hy-MT2-7B under the identical QLoRA recipe and found it reached a higher TVsub sacreBLEU than the 1.8B baseline, but at roughly $4.6\times$ the wall clock cost on the same hardware, and the zero shot comparison in Section 6.5 shows that fine-tuning the smaller model closes, and on this in-domain test split reverses, the gap to the larger model’s un-adapted translation quality. We did not fine-tune the 7B variant at full scale ourselves, both for this compute-cost reason and because it falls outside the scope stated in Sec-

tion 7, so we cannot say whether full-scale QLoRA fine-tuning would narrow, preserve, or widen the reduced-budget gap for the larger model.

As per the task description (WMT26 Organizers, 2026), all primary submissions to the Video Subtitle Translation task are judged with both automatic metrics and human evaluation, rather than by an automatic score alone. The exact automatic metric suite and the human evaluation protocol are not published on the task page as of this writing, which is one reason we present our TVsub numbers (Table 7) strictly as internal validation rather than as a prediction of official leaderboard performance. The task also distributes per-video meta-data (video description or synopsis, with the task page additionally referencing character lists and glossaries) and explicitly encourages participants to use it to resolve domain specific terminology and character names, since this is one of the properties the shared task is specifically designed to evaluate. Our decision not to condition on this meta-data at inference time (Section 5) is examined with reduced-budget evidence in Section 6.4 and plausibly costs us quality on exactly the cases the official evaluation is intended to reward (Section 8).

4.3 System Architecture

Figure 1 illustrates the training architecture.

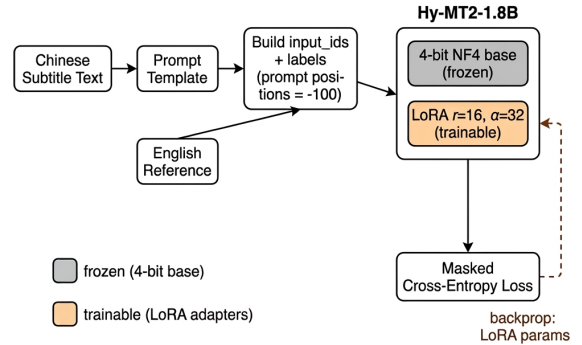


Figure 1: Training architecture of the submitted model.

The source subtitle line is embedded in the fixed instruction prompt (Section 4.4) and tokenized via the model’s chat template. The target translation is appended and only its tokens contribute to the loss. The frozen, 4-bit-quantized projection weights are augmented with a trainable low-rank LoRA branch (Section 4.5) inside each transformer block; gradients flow only through the LoRA pa-

rameters and the (dequantized, bf16) activations, never into the frozen 4-bit weights themselves.

4.4 Prompt Format

Both training and inference use the same single-turn instruction prompt, applied via the tokenizer’s chat template:

```
Translate the following text into
{tgt_lang}. Note that you must ONLY
output the translated result without
any additional explanation:{src}
```

with `tgt_lang` fixed to `English`. The target translation followed by an end-of-sequence token is appended after the prompt, and only target tokens contribute to the loss (prompt tokens are masked with label value -100) during training. The target is truncated from the end to fit if a prompt-plus-target sequence would exceed our maximum total length. The identical prompt, applied via each model’s own chat template, is used for the zero shot baselines in Section 6.5, so that the only differences between the baseline and fine-tuned conditions are the presence of the trained LoRA adapter and, where noted, the decoding configuration of Section 4.9.

4.5 LoRA Reparameterization

For each targeted projection matrix $W_0 \in \mathbb{R}^{d \times k}$ in the quantized base model, LoRA (Hu et al., 2021) adds a low-rank update

$$W = W_0 + \frac{\alpha}{r} BA, \quad (1)$$

where $B \in \mathbb{R}^{d \times r}$, $A \in \mathbb{R}^{r \times k}$, $r = 16$ is the adapter rank, and $\alpha = 32$ is the scaling factor, so that $\alpha/r = 2$. A is initialized from a Gaussian distribution and B is initialized to zero, so that $W = W_0$ at the start of training. For an input activation x , the forward pass through an adapted projection is

$$h = W_0 x + \frac{\alpha}{r} B (A \text{ drop}_{0.05}(x)), \quad (2)$$

with dropout applied to the LoRA branch’s input only, and W_0 frozen (stored 4-bit NF4, dequantized on the fly for the matrix multiply).

4.6 Training Objective

For a tokenized prompt-plus-target sequence of length T with target tokens occupying positions $p + 1, \dots, T$ (positions $1, \dots, p$ being the masked prompt), the model is trained to minimize the token averaged, masked causal language-modeling loss

$$\mathcal{L}(\theta_{\text{LoRA}}) = -\frac{1}{T-p} \sum_{i=p+1}^T \log P_{\theta}(y_i | y_{<i}), \quad (3)$$

where θ_{LoRA} denotes the trainable LoRA parameters $\{A, B\}$ across all targeted projections (the

base weights θ_{base} are held fixed), and $y_{<i}$ includes both the prompt tokens and any previously generated target tokens. Equivalently, every prompt-position label is set to -100 so that the standard cross-entropy loss ignores it, matching the masking in Section 4.4.

4.7 Learning Rate Schedule

The learning rate at optimizer step s follows a linear warmup for the first $s_w = \lceil 0.03 S \rceil$ steps followed by cosine decay over the remaining steps, where S is the total number of optimizer steps:

$$\eta(s) = \begin{cases} \eta_{\max} \frac{s}{s_w}, & s < s_w \\ \eta_{\max} \cdot \frac{1}{2} \left(1 + \cos \left(\pi \frac{s - s_w}{S - s_w} \right) \right), & s \geq s_w \end{cases} \quad (4)$$

with $\eta_{\max} = 2 \times 10^{-4}$. $S = \lceil N_{\text{batches}}/8 \rceil \times 3$ accounts for gradient accumulation over 8 steps and up to 3 epochs (fewer if early stopping triggers first).

4.8 Training Configuration

Table 1 summarizes our training configuration. We train for up to 3 epochs with early stopping (patience 2, minimum improvement 0.05 sacreBLEU) evaluated on the 500-pair development split at the end of each epoch, using gradient accumulation over 8 steps of batch size 4 (effective batch size 32). The optimization uses PagedAdamW8bit (Dettmers et al., 2021) with a cosine learning rate schedule and 3% linear warmup, gradient clipping at norm 1.0, and no weight decay. The training ran on a single NVIDIA L4 GPU (23.7GB) on a Lightning AI Studio instance.

Hyperparameter	Value
Base model	tencent/Hy-MT2-1.8B
Quantization	4-bit NF4 with double quantization
LoRA rank / α	16 / 32
LoRA dropout	0.05
Target modules	$q, k, v, o, \text{gate}, \text{up}, \text{down}$
Training pairs	60,000
Development pairs	500
Trainable / total parameters	19,398,656 / 1,810,479,104 (1.07%)
Maximum sequence length	640 tokens
Maximum epochs	3 (early stopping)
Batch size / Grad. accumulation	4 / 8
Optimizer	PagedAdamW8bit
Learning rate	2×10^{-4} (cosine)
Warmup ratio	0.03
Weight decay	0.0
Maximum gradient norm	1.0
Early stopping	Patience = 2, $\delta = 0.05$ sacreBLEU
Hardware	1 $\times$ NVIDIA L4 (23.7 GB VRAM)

Table 1: Training hyperparameters and optimization settings.

4.9 Decoding Strategy

We decode with beam search, no sampling, and a fixed token budget (Table 2) at both dev/test evaluation time and for the official test submission, which we found to be generous relative to typical subtitle line lengths. We apply a repetition penalty of 1.15 and a no-repeat n-gram size of 3 to reduce repetition collapse. This decoding configuration is held identical across every reported condition in this paper: dev/test evaluation of the final adapter, the zero shot baselines in Section 6.5, and official test inference (Section 5); the only exception is the generation length, which we ablate explicitly in Section 6.3 because several of our exploratory local runs used a shorter 40 token generation budget rather than the 160 token budget used for the final reported configuration. We treat decoding as an experimental variable rather than an implementation detail: Section 6.3 reports both the nominal effect of repetition penalty, no-repeat n-gram size, and generation length budget on the scores we obtain, and a paired bootstrap significance analysis of that effect given the size of our test set.

Generation Parameter	Value
Max new tokens	160
Beam size	4
Do sample	False
Temperature	1.0
Repetition penalty	1.15
No repeat n-gram size	3

Table 2: Decoding configuration, identical across dev/test evaluation, the zero shot baselines (Section 6.5), and official test inference. This is the single decoding strategy used for every number reported as “SubVision” in this paper.

5 Official Test Inference Pipeline

We run inference directly on the organizer-provided test files for the official shared-task submission. Each source file is a `*_zh.srt` subtitle file; we parse it into blocks of (index, timestamp, text) using a line-based SRT parser that tolerates blocks with or without subtitle text. Each block’s text is embedded in the training prompt template (Section 4) with target language fixed to English; blocks with empty source text are still passed through the model (with a placeholder single-space input, to keep batch shapes uniform), but their generated output is discarded and the block’s translation is forced back to an empty string, so empty subtitle timings are preserved without asking the model to hallucinate content. The batches of 16 blocks are decoded with the same beam search and generation settings as training time evaluation (Table 2). The first line of the decoded continuation is kept as the translation (guarding against rare cases

of trailing spurious continuation), and the result is written back into a new SRT file with the original index and timestamp preserved, so that timing synchronization from the source file is not altered.

Figure 2 below illustrates the pipeline. Critically, this pipeline does *not* incorporate the per-video metadata (title, description, character list, glossary) that the shared task makes available. Each subtitle line is translated in isolation, using only the fixed instruction prompt and the line’s own text, with no document level or cross-line context and no metadata conditioning (Sections 4.2, 8).

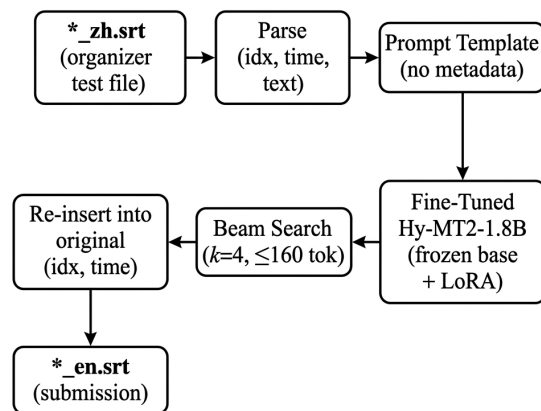


Figure 2: Test inference pipeline

6 Experiments

6.1 Setup

We evaluate on the TVsub test split (200 pairs with multi-reference SGM annotations, parsed as described in Section 3), reporting corpus-level BLEU (Papineni et al., 2002), sacreBLEU (Post, 2018), chrF (Popović, 2015), and TER (Snover et al., 2006), with generation performed under the fixed decoding configuration of Section 4.9 (Table 2). We report both BLEU and sacreBLEU because they are not directly comparable: BLEU uses our own tokenization, whereas sacreBLEU applies a fixed, standardized tokenizer (13a) specifically to make scores reproducible and comparable across papers. Therefore, we treat sacreBLEU as the primary metric for this reason and include raw BLEU only for continuity with the original TVsub benchmark protocol. All references for a given source segment are used jointly (multi-reference scoring), padding shorter reference lists to a common length as required by the scoring implementation. We additionally report a paired bootstrap significance

analysis given the modest size of this test split alongside the point estimates (Section 6.3.1), following standard practice for small sample MT evaluation.

6.2 Hyperparameter Tuning

We explored alternative settings including different LoRA ranks, learning rates, batch sizes, numbers of training pairs, numbers of training epochs, source/target/total sequence-length limits, and generation-length budgets, beyond the final configuration in Table 1. Section 6.3 reports these nine locally trained configurations directly, evaluated under the corrected SGM parser and, where noted, under two different decoding generation-length budgets, so that the nominal effect of decoding choice on the ranking is visible; Section 6.3.1 then tests how much of that ranking survives re-sampling uncertainty at this test set size.

6.3 Comparison Across Locally Trained Configurations

We trained nine further QLoRA configurations locally, varying training-pair count (60k vs. 100k), LoRA rank/ α (16/32 vs. 32/64 vs. 32/32), learning rate (5×10^{-5} – 2×10^{-4}), maximum sequence length (96 vs. 640 tokens), and batch/accumulation configuration, while holding the QLoRA target modules (q, k, v, o , gate, up, down), maximum gradient norm (1.0), checkpoint interval (every 500 steps), early-stopping patience (2), and the 500-pair (60k runs) or 800-pair (100k runs) development split fixed across all ten configurations. These nine configurations were run after, and partly to test, the findings of the reduced-budget ablation sweeps described in Section 6.4, which were conducted earlier in development. Table 11 (Appendix A) summarizes the ten training configurations, and Table 3 summarizes the decoding relevant evaluation settings used for each.

Sequence Length budget derivation. Runs R5–R9 use a substantially shorter length configuration (`max_src_len=56`, `max_tgt_len=40`, `max_total_len=96`, `eval_gen_max_new=40`) than the 640-token budget used for the final configuration. These four values were derived empirically, using two independent data sources cross-checked against each other. TVsub was used as a training proxy because no official WMT26 training data exists for this task, so its length distribution alone could not be trusted to reflect the real evaluation target. Therefore, prompt-token lengths were measured directly from the actual WMT26 demo and 100-video test set (41,837 real subtitle segments, tokenized with the model’s own tokenizer and wrapped in the exact same instruction template and chat-template scaffolding used at inference), giving a source-prompt length of $p_{99} = 37$

and $p_{999} = 42$ tokens under this task’s real conditions. This was then matched against TVsub’s own post-template prompt-length distribution (measured the same way, via `build_input_ids` over 10,000 sampled training pairs), which showed a p_{99} of 49 and a maximum of 55 tokens. TVsub’s fine-tuning sentences run slightly longer than the actual WMT26 test lines, so the training side cap needed to dominate rather than the target-side estimate. The setting `max_src_len=56` and `max_total_len=96` therefore covers the full observed TVsub source-length range (`max=55`) with the target evaluation distribution safely nested well inside it ($p_{999} = 42$), while `max_tgt_len=40` and `eval_gen_max_new=40` are sized from TVsub’s target-length p_{999} (23) and maximum (30) with comfortable headroom. The generation side cap only needs to bound how much the model is allowed to produce and is therefore decoupled from prompt length. This approach, grounding the choice in tokenized percentile statistics of both the training proxy and the real evaluation data rather than a round-number default, make sure under 1% of examples are truncated during fine-tuning while avoiding the unnecessary compute and memory cost of an oversized context window. The placeholder values of 256/256/640 used for the final configuration and R1–R4 would have been 4–7 $\times$ larger than what the data actually requires, a difference we deliberately keep in the comparison (Section 6.3) because it lets us measure what the shorter, empirically justified budget costs in practice against the final configuration’s more generous one.

Run	Max new	Beams	Rep. pen.	No-repeat n
R1	160	4	1.15	3
R2	160	4	1.15	3
R3	160	4	1.15	3
R4	160	4	1.15	3
R5	40	4	1.15	3
R6	40	4	1.15	3
R7	40	4	1.15	3
R8	40	4	1.15	3
R9	40	4	1.15	3
SubVision	160	4	1.15	3

Table 3: Decoding configurations used to evaluate the ten runs. Repetition penalty, beam width, and no-repeat 3-gram constraints are held fixed throughout; only the maximum number of newly generated tokens varies, with 40 tokens used for five exploratory runs (R5–R9) and 160 tokens used for the remaining five runs (R1–R4 and SubVision).

Table 4 reports test set metrics (200 TVsub pairs, multi-reference, sacreBLEU 13a) for all ten configurations under their respective decoding settings from Table 3. SubVision obtains the highest nominal sacreBLEU among the ten configurations, with R6 and R3 close behind (37.34 and 37.13 re-

spectively). Section 6.3.1 tests whether this ordering is statistically meaningful given the size of the test set; as summarized there, it is not: none of the pairwise comparisons involving SubVision reach significance at the 95% level.

Run	sBLEU	chrF	TER	BLEU
SubVision	37.63	52.76	53.93	38.56
R6	37.34	52.73	54.43	38.39
R3	37.13	52.49	54.61	37.92
R4	36.40	52.33	56.59	37.25
R7	36.36	52.54	56.78	37.21
R5	36.35	51.38	56.28	37.40
R1	35.46	51.21	57.27	36.44
R2	35.33	51.10	58.07	36.23
R9	34.98	50.90	57.58	35.91
R8	34.76	50.65	56.96	35.61

Table 4: Test set metrics for all ten configurations, sorted by sacreBLEU descending. Lower is better for TER. All rows use the identical 200-pair TV-sub test split, the corrected multi-reference SGM parser, and the decoding settings of Table 3.

6.3.1 Statistical Significance of the Run Comparison

We assessed how much of the observed ranking survives resampling uncertainty because Table 4 is computed on only 200 segments. For each of the ten configurations we drew 1000 paired bootstrap resamples of the test set (sampling segment indices with replacement, and recomputing corpus-level sacreBLEU on each resample using the same segments for every system in a given resample), giving a 95% confidence interval and a pairwise win rate for every one of the $\binom{10}{2} = 45$ system pairs.

Run	Boot. mean	95% CI	Std
SubVision	37.36	[33.10, 41.76]	2.17
R6	37.09	[33.09, 41.50]	2.14
R3	36.87	[32.81, 41.17]	2.13
R4	36.16	[31.58, 40.60]	2.35
R5	36.09	[31.96, 40.10]	2.17
R7	36.07	[32.03, 40.17]	2.10
R1	35.26	[31.14, 39.47]	2.17
R2	34.94	[31.12, 38.82]	2.02
R9	34.78	[30.71, 39.21]	2.12
R8	34.50	[30.36, 38.71]	2.14

Table 5: Bootstrap mean sacreBLEU and 95% confidence interval (1000 resamples) for each of the ten configurations, sorted by bootstrap mean descending. Only 1 of the 45 pairwise comparisons among these ten configurations has a 95% bootstrap confidence interval for the paired sacreBLEU difference that excludes zero, and it does not involve SubVision.

Table 5 reports the bootstrap mean and 95% CI for each configuration’s sacreBLEU. Across the full set of 45 pairwise comparisons, only one has a 95% bootstrap confidence interval for the paired sacreBLEU difference that excludes zero (R6 vs. R8); this pairing does not involve SubVision. Sub-

Vision’s difference in particular, from the lowest-scoring configuration in the table, R8, is nominally the largest gap observed (2.86 sacreBLEU) but is not significant (95% CI $[-0.52, 6.28]$, bootstrap win rate = 0.949). We therefore report SubVision’s ranking in Table 4 as the nominal top score among the ten configurations tested, rather than as a statistically established advantage over the other nine. We also tested the paper’s grouped decoding-budget claim directly: pooling the five 160-token-budget configurations (SubVision, R1–R4) against the five 40-token-budget configurations (R5–R9) under the same paired bootstrap procedure gives a mean difference of 0.41 sacreBLEU in favor of the 160-token group (95% CI $[-1.15, 2.10]$, bootstrap win rate = 0.691), which is not significant. The nominal ordering in Table 4, where the five 160-token configurations have a higher mean score than the five 40-token configurations, although R6 (40 tokens) ranks second and exceeds several of the 160-token configurations, is consistent with a real effect of generation budget, but at $n = 200$ we cannot distinguish it from sampling noise with this analysis, and we report it as directional rather than established.

We select SubVision as our reported and submitted configuration on the basis of its nominal top score in Table 4 and because it uses the longer, 160-token generation budget together with the 640-token training sequence length, which is directionally, though not significantly, the stronger combination among those tested; we do not claim this selection is supported by a statistically distinguishable advantage over the nearest alternatives (R6, R3).

6.4 Reduced Budget Ablations Justifying Design Choices

Four additional sweeps were run at a deliberately reduced training budget (fewer samples and epochs than a full run, so that a whole grid fits in a single session on 2×T4 GPUs) to decide, before committing full-scale compute, among alternative prompting strategies, hyperparameters and decoding settings, metadata-conditioning methods, and model/quantization configurations. These sweeps were conducted first, ahead of the ten full-budget training configurations reported in Section 6.3; the nine locally trained configurations were designed and run afterward, informed by these sweep results, including deliberately re-testing at full scale a setting the reduced-budget sweep had favored (LoRA rank 32/ $\alpha 64$, in R1 and R2) to check whether the reduced-budget advantage held once training reached convergence. Like the ten full-budget configurations and the zero shot baselines, all four sweeps were run against the corrected multi-reference SGM parser (Section 3), so every sacreBLEU figure reported in this section and in

Table 10 is post-fix and free of the reference conflation error described there. Each candidate was subject to a promotion gate that blocks a nominal TVsub sacreBLEU winner from being carried into the full-scale run unless it also (i) stays within a noise floor of 0.5 sacreBLEU over the instruction-prompt baseline (29.49 sacreBLEU), (ii) does not exceed a maximum repeated n-gram run under a decoding stress test, and (iii) renders a small set of anchor terms, 王妃 (*wángfēi*, “princess consort”), 仵作 (*wǔzuò*, “coroner”), and 王爷 (*wángyé*, “prince”), consistently across repeated occurrences in a held out episode. We treat these reduced budget results as a cheap screening pass rather than a confirmatory evaluation: candidates were filtered using the noise floor and consistency gate described above, run at a single reduced training scale with no re-sampling or significance testing, rather than with the bootstrap procedure of Section 6.3.1. A candidate passing this gate indicates it is worth the cost of full scale training, not that its nominal improvement is statistically established. Only the ten full-budget configurations of Section 6.3 receive that stronger evaluation. Table 10 (Appendix A) summarizes every candidate tested and the outcome. The paragraphs below give the reasoning behind each row.

Prompting strategy. Five templates were compared under matched fine-tuning and evaluation: the submission’s default single-turn instruction (`instruction_v1`), the same instruction moved into a system turn with a bare source string as the user turn (`bare_v1`), a minimal tag format with no natural-language instruction (`tagged_v1`), the default instruction preceded by one static in-context example (`fewshot_v1`), and an instruction-as-system-message variant (`system_role_v1`). `bare_v1` scored highest on test sacreBLEU but was blocked at the promotion gate (Table 10). We retained `instruction_v1` as the submitted prompt (Section 4.4): the nominal metric gain did not survive the consistency check, and instruction-format templates are also the most transparent to reason about and reproduce.

LoRA rank and decoding parameters. At the reduced budget, LoRA rank 32 ($\alpha = 64$) outperformed the submission’s rank 16 ($\alpha = 32$) and cleared the promotion gate, suggesting the larger adapter converges faster under a short schedule. We do not carry this result forward as-is: the full-budget comparison in Table 4 shows the opposite ordering once training reaches convergence, which we read as the reduced-budget rank advantage being a convergence-speed artifact of the short sweep rather than a capacity advantage that persists at full scale. We note that per Section 6.3.1, this reversal itself involves comparisons (R1/R2 vs. R3)

that were also not individually significant under the bootstrap test, so we treat the reduced-budget rank result as suggestive rather than as a confirmed reversal. A decoding-parameter grid over repetition penalty $\{1.0, 1.15, 1.30\}$ and no-repeat n-gram size $\{0, 3\}$ found no measurable repetition collapse at any setting under this specific stress test, but repetition penalty 1.30 began to alter translation content (Table 10). This is why SubVision uses repetition penalty 1.15 (Table 2): the point that still protects against repetition failures observed in production scale generation without the content drift seen at 1.30.

Metadata conditioning. Because TVsub carries no document boundaries in its training split and the released splits ship no real per-video titles or synopses, two document-context proxies were tested quantitatively on TVsub itself (Table 10), while genuine WMT26 title/synopsis metadata was tested only qualitatively on the official demo package, since it carries no reference translations to score. Both quantitative proxies were blocked or net-negative, and the one qualitative test possible with real metadata showed no observable effect at all on the small set of probes examined. We treat this qualitative result as a weak signal rather than a settled finding, since it reflects a handful of hand inspected outputs rather than a scored evaluation. A systematic test of metadata conditioning against real WMT26 metadata remains open (Section 8). These results together are why SubVision does not condition on per-video metadata for this submission (Sections 5, 8).

Model size and quantization. The fine-tuning of larger tencent/Hy-MT2-7B using the same QLoRA setup (still 4-bit NF4, LoRA r16/ α 32) produced a gain over the 1.8B baseline, but at roughly $4.6\times$ the wall clock cost on the same hardware (Table 10). The restriction of LoRA to attention only or MLP only projections underperformed adapting all targeted projections, confirming the choice to target all seven projection types in Table 1. An 8-bit quantization variant scored marginally higher than the 4-bit NF4 default at roughly double the memory footprint, and two “combined-best” configurations assembled from the individually best scoring settings above were both blocked at the promotion gate. This combination of findings is consistent with, and adds a reduced-budget data point to, the model-choice rationale already given in Section 4.2; as with the other reduced-budget results in this section, these deltas were not evaluated with the bootstrap procedure of Section 6.3.1 and should be read as screening evidence rather than confirmed effects. A larger base model or higher precision quantization can nominally raise the TVsub score, but neither survived the consistency gate and the more expensive combined configurations

did not compound into a net improvement.

6.5 Baseline Comparison

We evaluate two zero shot baselines: the same 4-bit NF4-quantized `tencent/Hy-MT2-1.8B` used in Section 4, but with no LoRA adapter attached, and the larger `tencent/Hy-MT2-7B` variant from the same model family, also 4-bit NF4-quantized and without any adapter to isolate what QLoRA fine-tuning contributes on top of the base model. The both baselines are decoded directly from the publicly released checkpoints, with no exposure to TVsub during training. We evaluate both on the identical 200 pair TVsub test split and multi-reference set used in Section 6, with the identical instruction prompt (Section 4.4) and the identical decoding configuration (Table 2), so that the fine-tuned adapter is the only variable that differs between these numbers and Table 7. Table 6 reports the results.

System	BLEU	sBLEU	chrF	TER
Hy-MT2-1.8B	11.80	11.60	38.32	72.96
Hy-MT2-7B	16.08	15.79	42.81	68.20
Hy-MT2-1.8B + QLoRA	38.56	37.63	52.76	53.93

Table 6: Zero shot NF4 baselines versus QLoRA fine-tuned Hy-MT2-1.8B on the identical 200 pair TVsub test split and decoding setup. Lower is better for TER.

The zero shot Hy-MT2-1.8B baseline trails our fine-tuned adapter by 26.03 sacreBLEU, 14.44 chrF, and 19.03 TER points. This gap is far larger than any of the pairwise differences examined in Section 6.3.1 among the ten fine-tuned configurations and we would expect it to remain significant under the same paired bootstrap procedure given its magnitude relative to the per configuration standard deviations in Table 5 (~ 2.1 – 2.4 sacreBLEU), though we did not run the baselines themselves through the bootstrap analysis. We take this gap as confirming that the gains we report in Table 7 come from the QLoRA fine-tuning step itself rather than from the base model’s pretrained translation ability alone. The un-adapted 1.8B model also underperforms the un-adapted 7B model by 4.19 sacreBLEU, which is expected given the roughly fourfold difference in parameter count; notably, our fine-tuned 1.8B adapter still exceeds the larger zero shot 7B model by a wide margin (37.63 vs. 15.79 sacreBLEU) which suggests that at least on this in-domain subtitle test split, domain specific QLoRA adaptation of the smaller model is considerably more effective than relying on the base model’s scale alone.

We did not fine-tune the 7B variant ourselves at full scale, both for the hardware reasons dis-

cussed in Section 4.2 and because it falls outside the scope stated in Section 7. Thus, we cannot say whether QLoRA fine-tuning would narrow, preserve, or widen this gap for the larger model. We treat the zero shot Hy-MT2-7B number here strictly as a scale reference point rather than as a competitive baseline system.

6.6 Results

Table 7 reports our final metrics, produced by loading the best early-stopped adapter and decoding the 200 pair TVsub test split with the configuration in Section 4. This test split is drawn from the same TVsub distribution and preprocessing pipeline as our training data and therefore serves as an in-domain diagnostic of whether fine-tuning succeeded, rather than as an estimate of performance on the shared task’s official, historical costume drama test videos, which come from a different source and distribution. We did not have access to references for the official test set and so could not compute an analogous score there. The reported numbers should be read strictly as an internal validation, and, per Section 6.3.1, as a nominal point estimate whose ranking against nearby configurations is not statistically established at this test set size.

Metric	Score
BLEU	38.56
sacreBLEU	37.63
chrF	52.76
TER	53.93

Table 7: Fine-tuned Hy-MT2-1.8B on the 200 pair TVsub test split, multi-reference scoring.

6.7 Sentence-Level Score Distribution

Table 7 reports corpus-level metrics, which can mask how consistently the model performs across individual segments. We additionally computed per-sentence sacreBLEU (13a tokenization) for all 200 test pairs (Table 8). The mean sentence-level score (43.43) is noticeably higher than the corpus-level sacreBLEU in Table 7 (37.63), which is expected: corpus BLEU aggregates n-gram counts across the whole set rather than averaging per-sentence ratios, so it is pulled down by low-scoring segments in a way a simple mean is not. The max of 100.0 reflects exact or near exact matches on short, common phrase subtitle lines (Table 12). Only 2 of 200 segments scored zero, both discussed in Section 6.8.

6.8 Qualitative Examples

Table 12 (Appendix A) presents representative best and worst predictions, ranked by per-sentence sacreBLEU, from the same 200 pair evaluation set.

Statistic	Value
Median sentence BLEU	33.73
Mean sentence BLEU	43.43
Max sentence BLEU	100.0
BLEU = 0 count	2 / 200

Table 8: Per-sentence sacreBLEU statistics on the 200 pair TVsub test split.

For each example, we show the Chinese source sentence with a Pinyin romanization, the model hypothesis, and the first available English reference (each segment has up to three references).

The highest scoring examples are short, common conversational utterances where the model closely matches the reference in both wording and meaning. The lowest-scoring examples in contrast illustrate well known limitations of reference based evaluation. Short idiomatic expressions (e.g., 讨厌啦, *tǎoyàn la*) and proper nouns (e.g., 安洁丽戏院, *Ānjiélì xìyuàn*) receive low sentence-level sacreBLEU despite producing fluent English, because the model selects different lexical realizations than the reference. Similarly, longer conversational sentences may preserve the intended meaning while differing substantially in wording, resulting in lower n-gram overlap. These examples highlight that BLEU family metrics can underestimate translation quality when acceptable paraphrases differ lexically from the available reference translations.

7 Experimental Scope

The primary objective of this paper is to present and validate a reproducible, resource efficient QLoRA fine-tuning pipeline for Hy-MT2-1.8B within the context of the WMT26 Video Subtitle Translation shared task, rather than to provide an extensive benchmarking study across different model scales, prompting strategies, metadata-conditioning approaches, or hyperparameter configurations. This work focuses on establishing the effectiveness and reproducibility of a single, validated training and decoding pipeline, together with zero shot baselines (Section 6.5) and a comparison against nine further locally trained configurations, including a bootstrap significance analysis of that comparison (Section 6.3), that contextualizes how much of the observed effectiveness is attributable to fine-tuning, training configuration, and decoding choice, rather than to the base model, its scale, or resampling noise at this test set size.

Several exploratory experiments were conducted during development. However, these results became invalid following the correction of the SGM multi-reference parsing error described in Section 3, which had corrupted the dev-set sacreBLEU used for early stopping and checkpoint selection as well

as the final test report. This rendered previous comparisons unreliable and necessitated complete retraining for any meaningful evaluation. We prioritized revalidating one end-to-end configuration using the corrected evaluation pipeline over re-running every exploratory configuration from scratch given the substantial computational cost of end-to-end fine-tuning under our available hardware constraints (a single 23.7GB GPU). Every run for which this paper reports a metric, SubVision itself, the nine locally trained configurations of Section 6.3, the zero shot baselines of Section 6.5, and the reduced-budget ablation sweeps of Section 6.4, was executed after this fix, against the corrected reference set, so none of them carry the invalidation concern that applies to the pre-fix exploratory numbers we chose not to report. The local configurations differ from SubVision in that they were trained and evaluated in a separate local environment, whereas SubVision additionally incorporates the SGM <DOC>-boundary offsetting fix at both the development-split scoring used for checkpoint selection and the final test scoring, which the nine local runs also apply post-fix but which was validated first in the external run.

Consequently, the experimental results presented in this paper are limited to the final, verified system (Sections 4–6.8), the nine locally trained comparison configurations and their bootstrap significance analysis (Section 6.3), and the two zero shot baselines described above. A comprehensive empirical investigation of alternative prompting strategies, metadata-conditioning methods, and length- or display-aware decoding constraints beyond repetition penalty and no-repeat n-gram size is left for future work.

8 Limitations

Limited evaluation size and domain coverage. Automatic evaluation is performed on 200 referenced TVsub segments drawn from the same general distribution as the training data. This makes the evaluation useful for validating the fine-tuning pipeline, but limits the strength of conclusions about performance on the official WMT26 test distribution. We cannot quantify how well the reported TVsub scores transfer to the shared task test data because reference translations for the official test data are not provided to participants.

No metadata conditioning. The submitted system does not use the per-video descriptions, character lists, or glossaries provided by the shared task. Such metadata could help resolve character names, domain specific terminology, and ambiguities that cannot be resolved from an individual subtitle line. Our reduced-budget exploratory evidence for metadata conditioning is based only on a small number of hand-inspected examples and

is therefore insufficient to establish whether meta-data would improve aggregate translation quality.

No cross line or document level context. Each subtitle line is translated independently. Consequently, the system does not explicitly enforce consistency of terminology, character names, or register across a video, and it cannot reliably resolve pronouns or ellipsis that require neighboring subtitle lines.

No length or display aware decoding. The decoding uses a fixed 160-token generation budget but does not explicitly optimize for characters per line, number of lines, reading speed, or other display constraints. We therefore evaluate translation quality primarily through reference based MT metrics rather than through a dedicated subtitle readability or display compliance evaluation.

Single decoding configuration per model. The same repetition penalty, no-repeat n-gram constraint, and beam size are used across the reported systems. This isolates the effect of the adapter and training configuration, but does not establish that the selected decoding configuration is optimal for every model or training setup.

Limited precision at $n = 200$. The paired bootstrap analysis with 200 test segments distinguishes only a small number of the pairwise differences among the ten trained configurations. The submitted configuration is not statistically distinguishable from the other nine. Accordingly, we interpret the choice of the submitted configuration primarily through its nominal score and engineering properties rather than as evidence of a statistically established optimum.

9 Conclusion

We presented SubVision, a QLoRA fine-tuning pipeline that adapts the 1.8B-parameter Hy-MT2 multilingual translation model to Chinese-English subtitle translation using the TVsub corpus, and the corresponding official test inference pipeline used to produce our WMT26 Video Subtitle Translation submission. The fine-tuned model on an in-domain TVsub test split reaches sacreBLEU 37.63, chrF 52.76, and TER 53.93 under a fixed decoding strategy (beam search, beam size 4, repetition penalty 1.15, no-repeat 3-gram constraint, 160-token generation budget), compared to zero shot NF4 baselines of 11.60 sacreBLEU for the un-adapted Hy-MT2-1.8B and 15.79 sacreBLEU for the larger un-adapted Hy-MT2-7B under identical prompt and decoding conditions. The large improvement over the un-adapted 1.8B model provides evidence that the observed gain comes from QLoRA fine-tuning rather than from the base model alone. We additionally compared SubVision

against nine further locally trained configurations that vary training pair count, LoRA rank, sequence length, and generation budget, all evaluated under the same decoding setting. A paired bootstrap analysis of this comparison shows that, at 200 test segments, SubVision’s nominal top ranking is not statistically distinguishable from several nearby configurations. We report this finding alongside the point estimates rather than presenting the ranking as an established result. Our shared-task submission applies the same fine-tuned model, prompt, and decoding configuration directly to the organizer-provided official test files, translating each subtitle line independently and without access to per-video metadata. We cannot report an automatic score for the actual submission because reference translations for the official test data are not provided to participants and we have described this gap explicitly rather than presenting the TVsub numbers as if they characterized official test performance. The next steps include incorporating the task’s provided per-video metadata into the prompt, adding cross-line context, evaluating length or display aware decoding constraints, and extending the statistical analysis in Section 6.3.1 to the reduced-budget ablation sweeps of Section 6.4.

Open Weights

The QLoRA adapter weights (`adapter_config.json`, `adapter_model.safetensors`; ~74 MB), tokenizer files, and chat template are publicly available under the Apache 2.0 license at <https://github.com/praaajg/wmt26-hymt2>. These are adapter only weights (LoRA rank 16) unlike a merged checkpoint, intended to be loaded on top of the frozen, 4-bit-quantized `tencent/Hy-MT2-1.8B` base model via a PEFT-compatible runtime, consistent with the training and inference pipeline described in Sections 4 and 5.

References

- Ankur Bapna and Orhan Firat. 2019. [Simple, scalable adaptation for neural machine translation](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1538–1548, Hong Kong, China. Association for Computational Linguistics.
- Chenhui Chu and Rui Wang. 2018. [A survey of domain adaptation for neural machine translation](#). In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1304–1319, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Tim Dettmers, Mike Lewis, Sam Shleifer, and Luke

- Zettlemoyer. 2021. [8-bit optimizers via block-wise quantization](#). *CoRR*, abs/2110.02861.
- Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. [Qlora: Efficient finetuning of quantized llms](#). *Preprint*, arXiv:2305.14314.
- Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. [Parameter-efficient transfer learning for NLP](#). In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. [Lora: Low-rank adaptation of large language models](#). *Preprint*, arXiv:2106.09685.
- Ahrii Kim. 2025. [Context is ubiquitous, but rarely changes judgments: Revisiting document-level MT evaluation](#). In *Proceedings of the Tenth Conference on Machine Translation*, pages 81–97, Suzhou, China. Association for Computational Linguistics.
- Yihang Li, Shuichiro Shimizu, Chenhui Chu, Sadao Kurohashi, and Wei Li. 2023. [Video-helpful multimodal machine translation](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4281–4299, Singapore. Association for Computational Linguistics.
- Sameen Maruf, Fahimeh Saleh, and Gholamreza Haffari. 2019. [A survey on document-level machine translation: Methods and evaluation](#). *CoRR*, abs/1912.08494.
- Evgeny Matusov, Patrick Wilken, and Yota Georgakopoulou. 2019. [Customizing neural machine translation for subtitling](#). In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 82–93, Florence, Italy. Association for Computational Linguistics.
- Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. [Training language models to follow instructions with human feedback](#). *Preprint*, arXiv:2203.02155.
- 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.
- Frithjof Petrick, Patrick Wilken, Evgeny Matusov, Nahuel Unai Roselló Beneitez, and Sarah Beranek. 2025. [AppTek’s automatic speech translation: Generating accurate and well-readable subtitles](#). In *Proceedings of the 22nd International Conference on Spoken Language Translation (IWSLT 2025)*, pages 222–231, Vienna, Austria (in-person and online). 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.
- Ashmari Pramodya, Yusuke Sakai, Justin Vasselli, Hidetaka Kamigaito, and Taro Watanabe. 2025. [Translating movie subtitles by large language models using movie-meta information](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pages 315–330, Vienna, Austria. Association for Computational Linguistics.
- Vikas Raunak, Roman Grundkiewicz, and Marcin Junczys-Dowmunt. 2024. [On instruction-finetuning neural machine translation models](#). In *Proceedings of the Ninth Conference on Machine Translation*, pages 1155–1166, Miami, Florida, USA. Association for Computational Linguistics.
- Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. 2006. [A study of translation edit rate with targeted human annotation](#). In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231, Cambridge, Massachusetts, USA. Association for Machine Translation in the Americas.
- Umar Baba Umar, Sulaimon Adebayo Bashir, and Abdulmalik Danlami Mohammed. 2026. [Anchoring the judge: Curriculum-based adaptation and reference-anchored MQM for LLM-based machine translation of an unseen low-resource language - a case of nupe](#). In *Proceedings of the Second Workshop on Language Models for Low-Resource Languages (LoResLM 2026)*, pages 200–211, Rabat, Morocco. Association for Computational Linguistics.
- Sebastian Vincent, Robert Flynn, and Carolina Scarton. 2023. [MTCue: Learning zero-shot control of extra-textual attributes by leveraging unstructured context in neural machine translation](#). In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8210–

8226, Toronto, Canada. Association for Computational Linguistics.

Longyue Wang, Zhaopeng Tu, Shuming Shi, Tong Zhang, Yvette Graham, and Qun Liu. 2018. [Translating pro-drop languages with reconstruction models](#). *CoRR*, abs/1801.03257.

Longyue Wang, Zhaopeng Tu, Andy Way, and Qun Liu. 2017. [Exploiting cross-sentence context for neural machine translation](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2826–2831, Copenhagen, Denmark. Association for Computational Linguistics.

WMT26 Organizers. 2026. [WMT26 video subtitle translation shared task](#). <https://www2.statmt.org/wmt26/video-subtitle-translation.html>.

Mao Zheng, Zheng Li, Tao Chen, Bo Lv, Mingrui Sun, Mingyang Song, Jinlong Song, Hong Huang, Decheng Wu, Hai Wang, Yifan Song, Yanfeng Chen, and Guanwei Zhang. 2026. [Hy-mt2: A family of fast, efficient and powerful multilingual translation models in the wild](#). *Preprint*, arXiv:2605.22064.

A Appendix

Idx	Source (Chinese)	Pinyin	English gloss
1	这鞠衣样式好生特别	Zhè jūyī yàngshì hǎoshēng tèbié	“This style of ceremonial robe is truly unusual.”
2	贵妃娘娘不如穿上	Guìfēi niángniang bùrú chuān shàng	“Consort, why not put it on.”
3	让我们开开眼	Ràng wǒmen kāikāi yǎn	“Let us see for ourselves.”
8	想要害婉儿的	Xiǎng yào hài Wǎn’ér de	“The one who wants to harm Wan’er.”
9	难道是曹嫔	Nándào shì Cáo pín	“Could it be Consort Cao?”
13	可以去紫堇花粉的毒性	Kěyǐ qù zǐjīn huāfěn de dúxìng	“It can remove the toxicity of corydalis pollen.”

Table 9: Selected source lines with Pinyin and English glosses added for accessibility for the official test demo file `b4101o4evls_zh.srt`

Sweep	Candidate	Δ sacreBLEU	Gate outcome	Reason
Prompting	bare_v1	+0.68	Blocked (printed verdict)	王爷 (<i>wángyé</i> , “prince”) rendered as “you’re dealing with the king of nine” and 仵作 (<i>wǔzuò</i> , “coroner”) as “a midwife who speaks for the living ...”; inconsistent renderings across near duplicate probes (“9th” vs. “ninth” prince; “zizzy” vs. “zee! zee!”)
Hyperparameters	Rank 32/ α 64 (reduced budget)	+0.80	Cleared, not adopted (printed verdict)	PROMOTE at $\Delta = +0.80$ (30.29 vs. 29.49); R3 (rank 16) outscores R1, R2 (rank 32) on every metric. Therefore, reversed at full scale.
	Repetition penalty 1.30	n/a	Not adopted	No repetition collapse detected (stress grid passed on r32 adapter), but degrades content: bracketed gloss “(xian’s tianlan sect)” became unglossed “(xianhuangzong)” at penalty 1.30.
Metadata	Previous-line context	+0.13	Blocked (printed verdict)	$\Delta = +0.13$ (29.62 vs. 29.49) below 0.5 floor; still rendered 王妃 (<i>wángfēi</i> , “princess consort”) / 仵作 (<i>wǔzuò</i> , “coroner”) inconsistently (3 distinct renderings each)
	[Episode N] boundary tag	−2.39	Blocked (no gate run)	Net negative vs. baseline ($\Delta = -2.39$: 27.10 vs. 29.49), judged on metric alone
	Real title/synopsis (qualitative)	n/a	No effect	Byte identical output to no-metadata baseline on all 3 probes tested
Model / quantization	Hy-MT2-7B	+2.78	Cleared, not adopted (no gate run)	Real gain ($\Delta = +2.78$: 32.27 vs. 29.49) but $\sim 4.6\times$ wall clock cost (282.6 vs. 61.4 min)
	Attention-only LoRA	−1.11	Not adopted	Underperforms all-target-module LoRA (28.38 vs. 29.49)
	MLP-only LoRA	−0.23	Not adopted	Underperforms all-target-module LoRA (29.26 vs. 29.49)
	8-bit quantization	+0.63	Not adopted	$\sim 2\times$ base-weight memory for marginal gain (+0.63: 30.12 vs. 29.49)
	Combined-best variants (2)	below floor	Blocked (printed verdict)	Inconsistent anchor term translations (3 distinct renderings each; $\Delta = -0.49$ / -0.28 : 29.00 / 29.21 vs. 29.49)

Table 10: Every candidate tested in the reduced-budget ablations, its nominal sacreBLEU delta against the 29.49 instruction prompt baseline, and its promotion gate outcome. “Cleared, not adopted” means the candidate passed the gate’s numeric and consistency checks but was rejected on other grounds (compute cost, or reversal at full scale). Outcomes marked “printed verdict” are verbatim `promote_or_block` outputs from the sweep notebooks. The remaining outcomes apply the same numeric and consistency checks without a gate run.

Run	Train	Val	Max total len	Batch/accum	r/α	LR / sched	Weight decay
R1	60k	500	640	8/4	32/64	2e-4 cosine	0.0
R2	100k	800	640	8/4	32/64	2e-4 cosine	0.0
R3	60k	500	640	8/4	16/32	2e-4 cosine	0.0
R4	60k	500	640	8/4	32/64	1e-4 cosine	0.01
R5	60k	500	96	8/4	32/64	1e-4 cosine	0.01
R6	100k	800	96	16/2	32/64	1e-4 cosine	0.01
R7	100k	800	96	16/2	32/32	5e-5 cosine	0.01
R8	100k	800	96	16/2	16/32	2e-4 cosine	0.0
R9	60k	500	96	16/2	16/32	2e-4 cosine	0.0
SubVision	60k	500	640	4/8	16/32	2e-4 cosine	0.0

Table 11: Training configurations for the nine locally trained runs (R1–R9) and the submitted system **SubVision**. All runs share QLoRA target modules (q, k, v, o , gate, up, down), gradient-norm clipping at 1.0, and early-stopping patience 2; “Max total len” is the combined prompt+target token budget, and “Batch/accum” is per step batch size over gradient accumulation steps.

Source	Pinyin	Hypothesis	Reference
<i>Highest-scoring</i>			
罗斯, 谢谢	Luósī, xièxiè	Thanks, Ross.	Ross, thank you.
不要告诉瑞秋, 待会见	Bùyào gàosù Ruìqiū, dāihuì jiàn	Don't tell Rachel. I'll see you later.	Don't tell Rachel. See you later.
你有口香糖吗?	Nǐ yǒu kǒuxiāngtáng ma?	Do you have any gum?	Do you have any gum?
<i>Lowest-scoring</i>			
讨厌啦	Tǎoyàn la	Oh, crap.	This sucks!
安洁丽戏院, 开车	Ānjiélì xìyuàn, kāichē	Cinema Angela, drive.	The Angelika! Go! Go!
因为我一点也不爱	Yīnwèi wǒ yìdiǎn yě	Because I don't love her. I	She means nothing to me.
她一点也不爱	bú ài tā, yìdiǎn yě bú ài	don't love her at all.	Nothing!

Table 12: Predictions from the 200 sentence TVSub evaluation set ranked by per-sentence sacreBLEU with Pinyin added for the Chinese source.