Multilingual Long-Form Speech Instruction Following: KIT's Submission to IWSLT 2026

Enes Yavuz Ugan[1], **Maike Züfle**[1], **Yuka Ko**[1], **Supriti Sinhamahapatra**[1], **Fabian Retkowski**[1], **Seymanur Akti**[1], **Jan Niehues**[1], **Alexander Waibel**[1,2]
[1]Karlsruhe Institute of Technology
[2]Carnegie Mellon University
firstname.lastname@kit.edu

Abstract

With the advent of Large Language Models, single-task and token-based multi-task models have evolved into instruction-based systems that infer task and target language implicitly from natural language prompts. This trend is reflected in IWSLT's Instruction Following Track, which this year introduced new tasks including an unknown surprise task, posing a genuine challenge against overfitting to known tasks. We present KIT's submission to the Long and Short Instruction Following tracks in the unconstrained setting. Our approach combines a general data augmentation pipeline that converts short-form corpora into long-form training data through segment concatenation, LLM-based label generation, and cross-lingual translation, yielding over 1M instances across six tasks and four languages. We further show that likelihood-based re-ranking, while highly effective for ASR, systematically degrades semantic tasks by spuriously selecting candidates generated from segmented audio processing rather than holistic long-form inference, a failure mode resolved by combining likelihood with Minimum Bayes Risk decoding.

1 Introduction

Recent work has focused on integrating speech into LLMs to create Speech Language Models, typically by incorporating pre-trained audio encoders (Tang et al., 2024; Koneru et al., 2025; Retkowski et al., 2025; Züfle and Niehues, 2025) into the LLM architecture. Alternative approaches train audio and vision encoders jointly with an LLM backbone, developing true multimodal Foundation Models (Xu et al., 2025a), with recent work also demonstrating the effectiveness of combining speech and vision modalities (Sinhamahapatra and Niehues, 2025; Koneru et al., 2026). Despite these advances, a significant gap remains: the effective processing of long-form audio (Papi et al., 2026). Most models rely on the Whisper encoder (Radford et al., 2022), which natively supports only 30 seconds per inference pass. While newer models such as Phi-4 (Abdin et al., 2024) and Qwen2.5-Omni (Xu et al., 2025b) remove these architectural constraints, they lack exposure to long-form audio during training, a gap that manifests as significant performance degradation even on basic ASR (Papi et al., 2026). This paper presents KIT's submission to the Unconstrained Long & Short Instruction Following tracks of IWSLT 2026. We participate in all four target languages: German, English, Italian, and Chinese. Our approach combines data augmentation to extend short-form datasets to long-form settings, temperature-scaled interleaving to balance task representation, and re-ranking to improve generation quality, exploring both end-to-end and cascaded architectures. The main contributions of this work are:

- A data augmentation framework that includes conversion of short-form speech datasets into long-form instruction-following training data through segment concatenation, LLM-based label generation, and cross-lingual reference translation, yielding a publicly released dataset of over 1M instances across six tasks and four languages.[1]
- An empirical comparison of fixed-probability and temperature-scaled data interleaving strategies, identifying $T = 2$ as a strong choice for multimodal speech instruction following.
- A negative result on Chain-of-Thought task-token conditioning, showing that prefix-based task routing fails under task imbalance and task similarity, leading to a collapse in task discrimination.
- A systematic comparison of six re-ranking

[1] YapayNet/iwslt2026-if-augmented

Proceedings of the 23rd International Conference on Spoken Language Translation (IWSLT 2026), pages 132–149

strategies under the realistic constraint of no task identity at inference time, revealing a previously uncharacterized failure mode where likelihood-based re-ranking spuriously selects segmentation-based candidates for semantic tasks.

- A combined Likelihood+MBR re-ranking strategy that resolves the ASR-vs-semantics tradeoff, achieving strong ASR improvement while limiting degradation on QA and summarization.

2 Data

This year's IWSLT instruction-following task covered six tasks: Automatic Speech Recognition (ASR), Speech Translation (ST), Spoken Question Answering (SQA), Speech Summarization (SSUM), Audio Chaptering (ACHAP), and a surprise task. All tasks except ASR support the language pairs en–en, de, it, zh; however, not all task–language combinations have in-domain data readily available. A further challenge is data format: most existing datasets consist of short utterances under 30 seconds, while the long-form track requires audio up to 15 minutes. We address both gaps through a three-stage augmentation framework: (1) segment concatenation with speaker-aware grouping to construct long-form audio, (2) LLM-based label generation to create task annotations for unlabeled or partially annotated data, and (3) cross-lingual reference translation to extend English-only annotations to all target languages. The resulting corpus contains over 1M training instances spanning all six tasks and four language pairs, summarized in Table 1.

2.1 Translation Augmentation

Most datasets provide annotations only in English. To cover all target languages (de, it, zh), we translate English references using `translategemma-12b-it`[2] (Team et al., 2025).

We select this model based on strong reference-free translation quality (COMETKiwi (Rei et al., 2022)), outperforming alternatives such as SeamlessM4T-Large (Barrault et al., 2023) and LLaMA-3.1-8B-Instruct (Kassianik et al., 2025). This procedure is applied consistently across tasks for language coverage augmentation.

2.2 ASR

Our original ASR datasets consist of short audio-transcript pairs, typically under 15 seconds. However, the long-form track requires audio up to 15 minutes, necessitating enhancements to generate training data aligned with this evaluation setting.

YTSeg. We repurpose YTSeg (Retkowski and Waibel, 2024), a dataset originally curated for chaptering, as a source of long-form video for ASR. We retain videos up to 15 minutes, reducing the dataset from 16,404 to 10,729 examples (34.6% reduction). We further filter videos where Whisper Large achieves $\geq$50% WER to remove noisy examples, yielding 10,638 examples (0.86% additional reduction). We then applied lightweight text normalization: removing non-lexical metadata such as background-noise markers in parentheses or square brackets (e.g., [music], (applause)), while preserving bracketed numeric content. We also normalized whitespace and excessive line breaks. This produced cleaner transcripts that better reflect spoken content for ASR training and evaluation.

NUTSHELL. We use NUTSHELL (Züfle et al., 2025), a dataset of ACL talk videos, which is particularly in-domain for our setting. Since NUTSHELL lacks ASR transcripts, we generate them using `parakeet-tdt-0.6b-v2`[3], applied to the audio track of each video. To align with the shared task's 15-minute duration limit, we filter out videos exceeding this threshold.

EuroParl. We extend EuroParl-ST (Iranzo-Sánchez et al., 2020) for long-form ASR by concatenating aligned speech segments into 5–10 minute chunks. Segments are grouped by session and speaker, with additional speakers included if needed. Only transcribed audio portions are concatenated, excluding untranscribed gaps, yielding clean long-form audio-transcript pairs.

LibriSpeech. We extend LibriSpeech (Panayotov et al., 2015) by combining `train-clean-360` and `train-other-500` to increase acoustic diversity. Utterances are grouped by chapter and shuffled at the chapter level to mix conditions while preserving intra-chapter order. Chapters are then greedily concatenated into segments of up to 10 minutes, yielding long-form audio–text pairs aligned with the evaluation setting.

[2] google/translategemma-12b-it

[3] nvidia/parakeet-tdt-0.6b-v2

We further apply truecasing and punctuation restoration using NVIDIA NeMo's (Kuchaiev et al., 2019) *PunctuationCapitalizationModel* (punctuation_en_bert), a BERT-based model (Devlin et al., 2019), converting lowercase, unpunctuated transcripts into well-formed text.

Collectively, these augmentations transform four short-form ASR corpora into long-form training data without requiring new recordings or manual annotation, demonstrating that existing resources can be effectively repurposed for long-form settings through concatenation and normalization alone.

2.3 Speech Translation (ST)

We use four datasets for the ST task:

EuroParl-ST. We apply the same long-form concatenation pipeline used for ASR to EuroParl-ST, generating long-form speech translations for the language pairs en–de and en–it.

CoVost. We apply a similar long-form concatenation pipeline to CoVost (Wang et al., 2020) to generate long-form speech translation pairs for en–zh.

NUTSHELL & LibriSpeech. We leverage respective transcripts and translate them to German, Italian, and Chinese.

2.4 Spoken Question Answering (SQA)

We rely on two SQA benchmarks and their translations:

LibriSQA. We use the open-form question subset of LibriSQA (Zhao et al., 2023), a spoken question-answering dataset built on LibriSpeech (Panayotov et al., 2015). We translate questions and answers into German, Italian, and Chinese as described earlier.

NUTSHELL. To address the lack of academic-domain coverage in LibriSQA, we adapt NUTSHELL for SQA. Using ASR transcripts, we prompt `gemma-3-12b-it`[4] (Team et al., 2025) to generate five questions per transcript: four answerable from the content and one unanswerable.

2.5 Multiple Choice (MC)

To improve generalization and mitigate overfitting and catastrophic forgetting, as well as to prepare for the surprise task, we incorporate multiple-choice style data.

LibriSQA (LibriMC). We use the multiple-choice subset of LibriSQA (Zhao et al., 2023) and translate questions into the missing three target languages.

MMSU. We leverage the Massive Multi-task Spoken Language Understanding and Reasoning Benchmark (MMSU) (Wang et al., 2025), using its multiple-choice questions to strengthen instruction-following capabilities.

2.6 Speech Summarization (SSUM)

We use the following two datasets for SSUM, and their translations into the other target languages.

NUTSHELL. We use NUTSHELL (Züfle et al., 2025), which pairs ACL talk videos with corresponding paper abstracts. We treat abstracts as reference summaries.

YTSeg. To increase training data for the SSUM task, we augment YTSeg with synthetic abstract-like summaries. We restrict the pool to talks categorized as *Science*, *Technology*, or *Education*, according to the LLaMA 2-based topic categories from Retkowski and Waibel (2024), to better match the academic domain of ACL test data. This filtering step leaves 5,700 (or 53.6%) of the 10,638 videos obtained from the preprocessing in Section 2.2 for training. We find that abstracts in NUTSHELL average approximately 145 words in length. We use this as an explicit length target when prompting `Qwen3.5-27B`[5] with the reference transcript, and three randomly sampled NUTSHELL abstracts as in-context examples to generate an abstract-like summary of comparable length and style for each selected talk.

2.7 Audio Chaptering (ACHAP)

Finally, for audio chaptering, we include two datasets, both of which are translated into all target languages:

YTSeg. For ACHAP, we utilize YTSeg and follow the same preprocessing as described in Section 2.2. We frame chaptering as a structured transcription task involving joint transcription, segmentation, and title generation, effectively extending the long-form ASR task. We use Markdown formatting for the structured output.

[4] google/gemma-3-12b-it

[5] Qwen/Qwen3.5-27B

Dataset	# Samples	Initial (%)	p (%, T=2.0)
ASR	19,248	1.84	6.12
SQA	474,888	45.31	30.41
MC	380,056	36.26	27.21
SSUM	35,748	3.41	8.34
ST	29,343	2.80	7.56
AChap	37,862	3.61	8.59
Instruct	71,013	6.78	11.76
Total	1,048,158	100.00	100.00

Table 1: Training data distribution with original proportions (init) and temperature-smoothed sampling probabilities ($T = 2.0$, where $p_i \propto n_i^{1/T}$).

NUTSHELL. Secondly, we augment NUTSHELL with chapter annotations generated by `Qwen3-Omni`[6], using the inference settings and ICL prompt from Retkowski et al. (2026), which demonstrated reasonable zero-shot performance on shorter ($\leq$20 min) single-speaker audio. We retain only samples of 3–15 minutes with WER $< 30\%$ against Parakeet-Transcript and 4–11 generated chapters.

2.8 General Instruction Following

To further enhance generalization and preserve instruction-following capabilities across domains, we incorporate general instruction-following data.

TowerBlocks. We use TowerBlocks (Alves et al., 2024) to construct an augmented multimodal corpus for instruction-following fine-tuning. TowerBlocks encompasses diverse subtasks including translation, named-entity recognition, post-editing, and paraphrase generation. We focus on the UltraChat subset, which provides a context-instruction format. We first filter samples matching this structure using Llama-3.1-8B [7], tagging corresponding components. We then generate speech samples from the context using Kokoro-82M [8], a state-of-the-art TTS model. This enables the model to refer to the speech during instruction-following fine-tuning.

3 Model Submissions

We submit two systems: an end-to-end model as our primary submission and a cascaded model as a contrastive system to analyze the trade-offs between the two approaches.

3.1 End-To-End Model

We select `Qwen2.5-Omni`[9] (Xu et al., 2025a) as our primary end-to-end model and use LLamaFactory (Zheng et al., 2024) as the training framework. We chose this model for its strong multimodal instruction-following capabilities. Due to time and hardware constraints, we were unable to evaluate the more recent Qwen3-Omni (Xu et al., 2025c).

3.2 Cascaded Model

We additionally explore a cascaded pipeline that decomposes the task into ASR followed by text-based instruction following. For ASR, we use `parakeet-tdt-0.6b-v2`[10], which generates transcripts from audio. These are passed to `Qwen2.5-7B-Instruct`[11] together with task-specific prompts to produce final outputs.

To match the shared task format, we replace the `<audio>` field with a `Transcript:` prefix followed by the ASR transcript, enabling fully text-based processing. Unless stated otherwise, we use the same training data and hyperparameters as in the end-to-end setup. For MMSU, transcripts are generated with `parakeet-tdt-0.6b-v2`. Inference uses greedy decoding (beam size 1) with a maximum length of 4096 tokens.

4 Experimental Setup

We now detail our training recipe, covering prompt design, hyperparameters, data interleaving strategies, and evaluation criteria.

4.1 Training Strategy

To ensure our model strictly adheres to given instructions, we employ a fixed and restrictive system prompt throughout training and inference. For better instruction generalization, we create several different instruction prompts per task and per language, and each training and development instance is randomly assigned one of these prompts. The system prompt and instruction templates are provided in Tab. A1 in the Appendix. We keep the system prompt fixed because it defines a consistent global behavior across all tasks. In contrast, task-specific instructions are varied to better approximate realistic user interactions, where the same task can be expressed through many different prompt

[6] Qwen/Qwen3-Omni-30B-A3B-Instruct
[7] meta-llama/Llama-3.1-8B
[8] hexgrad/Kokoro-82M
[9] Qwen/Qwen2.5-Omni-7B
[10] nvidia/parakeet-tdt-0.6b-v2
[11] Qwen/Qwen2.5-7B-Instruct

formulations. We hypothesize that prompt variation reduces template memorization and improves instruction-following robustness under prompt distribution shifts.

We also experiment with Chain-of-Thought (CoT) style predictions by introducing special task tokens with noise initialization while keeping all other settings unchanged. The task tokens include <|asr|>, <|st|>, <|mc|>, <|sqa|>, <|achap|>, <|ssum|>, and <|instruct|>, along with language tokens <|en|>, <|de|>, <|it|>, and <|zh|>. For SQA, we additionally include answerability tokens <|answerable|> and <|unanswerable|>. The token order follows: task token, language token, answerability token (if applicable), and then the target output.

4.2 Data Preparation

Our training data exhibits significant class imbalance across tasks (see Tab. 1). To ensure adequate task visibility during training, we maintain datasets separately and apply interleaving strategies. We explore two approaches: (1)a manually specified fixed-probability sampling strategy designed to partially compensate for task imbalance and redistribute sampling probability from less important tasks to more important ones, and (2) temperature-scaled sampling, where probabilities are derived from dataset sizes. We do not aim to exhaustively evaluate all possible fixed-probability sampling strategies, but rather to compare a reasonable heuristic sampling distribution with a principled temperature-scaled alternative.

For fixed-probability sampling, we assign the following probabilities: ASR (10%), SQA (31%), MC (9%), SSUM (14%), ST (13%), ACHAP (14%), and general instruction following (9%). The probabilities were selected heuristically based on dataset size and perceived task importance. In particular, we reduced the MC sampling probability because MC was treated as an auxiliary task rather than a primary evaluation target.

For temperature-scaled sampling, let n_i denote the number of training instances in dataset i and $N = \sum_i n_i$ the total number of instances. We obtain interleaving probabilities as:

$$p_i = \frac{n_i^{1/T}}{\sum_j n_j^{1/T}}, \tag{1}$$

with $T > 0$ interpolating between size-proportional sampling ($T = 1$) and uniform sampling ($T \to \infty$). We adopt $T = 2$, following prior work showing improvements on the ST task (Li et al., 2025), which corresponds to sampling proportional to the square root of dataset size ($p_i \propto \sqrt{n_i}$).

4.3 Hyperparameters

We experimented with various training configurations and found that an effective batch size of 4 with a learning rate of 1.0e-4 yielded the best results, outperforming configurations with larger batch sizes and learning rates ranging from 1.0e-5 to 2.0e-4. To prevent out-of-memory errors, we set the token cutoff length to 28,000 tokens. Since 15 minutes of audio requires approximately 22,500 tokens, this provides sufficient headroom for model predictions. We set the warmup ratio to 0.1 with a total of 60,000 update steps, and apply LoRA (Pham et al., 2021; Hu et al., 2022) with rank 32.

4.4 Evaluation

For model selection, we evaluate on the MCIF (Multimodal Crosslingual Instruction-Following) benchmark (Papi et al., 2026), YTSeg, and LibriMC. We additionally include an unused YTSeg validation subset with ACHAP translations for German, Italian, and Chinese. During training, we extract a small subset from each development split to monitor evaluation loss and inform checkpoint selection. We apply the following evaluation protocols per task:

- **ASR, ST, SQA, SSUM**: We follow the automated evaluation protocols of Papi et al. (2026).
- **ACHAP**: We follow Retkowski et al. (2026), employing Collar-F1 (with ± 3 second tolerance) to assess segmentation quality and the Global Concatenation protocol with BERTScore to measure title quality, as implemented in the `chunkseg` package[12].
- **MC**: We compute accuracy as $\frac{1}{N}\sum_{i=1}^{N} \mathbf{1}[\hat{y}_i = y_i]$, where predictions are normalized by stripping whitespace and must exactly match one of the choices {A, B, C, D}.
- **Surprise Task**: As the nature of the surprise task is unknown at training time, we train the model to identify the most similar known task and apply the corresponding behavior.

[12] https://github.com/retkowski/chunkseg

4.5 Results

We report results on the MCIF long-form track as well as on the ACHAP and MC tasks, which are evaluated separately due to their distinct evaluation protocols.

4.5.1 MCIF Results

Tab. 2 presents results on the MCIF long-form track with fixed and mixed prompts and beam size 1. We compare two baselines: Qwen2.5-Omni (end-to-end) and Cascaded (Section 3.2).

Baselines. The cascaded baseline demonstrates the advantage of using an ASR model specifically optimized for long-form audio, achieving 5.88% WER and 80.81% COMET score on translation. In contrast, the end-to-end baseline outperforms the cascaded approach on SQA and SSUM, but its inferior ASR and ST performance (53.40% WER, 68.65% COMET) reveals that it was not specifically designed for long-form audio.

Fine-tuning. Fine-tuning substantially improves both systems. We observe faster convergence and better final results with temperature-scaled data sampling (row 4) compared to our manually specified fixed-probability sampling strategy (row 3), and therefore use it for all subsequent experiments.

Temperature-scaled sampling with $T = 2$ consistently performs well across tasks (Table 2, rows 3 vs. 4). As $T = 2$ corresponds to sampling proportional to $\sqrt{n_i}$, this extends observations of (Li et al., 2025) from text-only to multimodal speech instruction following. This suggests that square-root proportional sampling is a strong default for multimodal speech instruction-following training.

In-Domain Adaptation and Checkpoint Averaging. Continuing training on NUTSHELL, an in-domain dataset of scientific talks similar to the evaluation data, yields clear improvements (row 5), supporting our hypothesis that in-domain adaptation benefits long-form performance. Checkpoint averaging of the best two checkpoints (row 6) provides further gains, whereas averaging additional checkpoints degraded performance. Since performance on Italian remained suboptimal (see Table A9), we additionally fine-tuned row (4) with Italian in-domain data and averaged the best checkpoint with those from row (6), yielding row (7). This model shows modest improvements on SSUM and ST tasks.

Segmented Evaluation. To diagnose whether poor ASR results stem from task confusion rather than fundamental ASR limitations, we applied segmented evaluation (SHAS) to row (7). The degradation on SQA and SSUM is expected, but acceptable ASR performance suggests that long-form nature, rather than the task itself, drives the poor unsegmented results.

Chain-of-Thought Conditioning. We evaluate CoT-style task-token conditioning (row 9) (Koneru et al., 2025), but observe degraded performance, especially for ASR (79.24% WER).

Task prediction collapses: ASR inputs are almost always misclassified as SSUM, despite only a slight data imbalance (~2%). Explicit ASR predictions are rare (3 fixed, 2 mixed) and occur only when the ground truth is SQA, indicating a failure to learn task discrimination.

We hypothesize that the model defaults to SSUM because it is both slightly more frequent and structurally closer to ASR (i.e., grounded in transcript-level content), making it an easier fallback. This suggests that prefix-based task routing remains weakly grounded and prone to shortcut behavior, requiring stronger balancing or supervision to be effective.

Cascaded System. Fine-tuning the cascaded system substantially improves performance on SQA (27.48 → 33.36%) and SSUM (13.31 → 28.55%) while preserving strong ASR performance (5.88 → 5.90% WER). The cascaded model achieves the best overall ST result (83.72% COMET) among all systems. These results highlight that the cascaded approach benefits from high-quality intermediate transcripts and excels at transcription-sensitive tasks. However, performance on semantic tasks such as SQA and SSUM lags the end-to-end model, likely due to error propagation from ASR transcripts and loss of audio-specific information. Additionally, training data aligned with the end-to-end setup may not be optimal for text-based instruction following, potentially limiting the cascaded model's ability to handle semantic reasoning tasks.

Model Selection. Model selection was based on robustness across all tasks. Row (7) consistently ranks as the second or third best performer across metrics, closely following the top model. For the surprise task, we hypothesized that the Italian fine-tuning checkpoint, when averaged with

Models	SQA (↑)		SSUM (↑)		ASR (↓)		ST (↑)	
	Fix.	Mix.	Fix.	Mix.	Fix.	Mix.	Fix.	Mix.
Baselines								
(1) Qwen 2.5 Omni	30.78	32.94	14.21	17.87	53.40	35.35	68.65	70.79
(2) Cascaded	27.48	27.66	13.31	13.40	**5.88**	**6.85**	80.81	*80.46*
Fine-tuned								
(3) Fixed-Prob. Sampling	36.16	36.42	28.77	28.76	30.59	33.61	75.58	74.90
(4) Temp. Sampling ($T = 2$)	37.75	37.87	28.87	28.99	27.58	38.43	76.10	75.96
(5) N2	*39.98*	*39.98*	**29.05**	28.75	25.98	30.46	74.98	74.94
(6) N2+Avg	**40.68**	**40.86**	26.06	*29.33*	29.36	34.48	73.35	75.42
(7) N2+IT+Avg (**Primary**)	40.42	40.31	*26.10*	29.41	37.65	36.88	73.94	76.01
(8) N2+IT+SHAS	-11.09	-12.14	2.93	5.90	*11.72*	*11.25*	*80.62*	80.84
(9) Chain-of-Thought (CoT)	34.88	35.50	26.56	**29.74**	79.24	80.04	42.09	43.69
(10) Cascaded+FT (**Contrastive**)	33.36	32.09	28.55	27.33	5.90	9.76	**83.72**	**83.75**

Legend: (5) continued fine-tuning on in-domain (NUTSHELL) data; (6) checkpoint averaging of (5); (7) additional Italian in-domain fine-tuning with averaging; (10) cascaded system with fine-tuning.

Table 2: MCIF long track comparison of fixed vs. mixed prompt evaluation (macro-averaged). Best results are in **bold**, second-best are underlined and third-best in *italic*. Shaded rows indicate submitted models.

others, could mitigate catastrophic forgetting and improve generalization (Ugan et al., 2025). Thus, we selected row (7) as our primary submission and row (10) as a contrastive system. Although the cascaded+FT model underperforms on SQA and SSUM, its superior ASR and ST performance and qualitative analysis suggest it may handle challenging surprise task instances.

4.5.2 ACHAP and MC Results

Results on ACHAP and MC tasks are presented in Table A11 in the Appendix. We evaluated MC only on English, as we identified it as a potential surprise task and reserved other languages for evaluation robustness. For ACHAP, we used 50 unused samples from the YTSeg development split, translating them into German, Italian, and Chinese for multilingual evaluation.

Both fine-tuned models substantially outperform the baseline on MC, achieving over 81% accuracy. For ACHAP, multilingual training yields large gains over the baseline in English, improving F1 from 1.20 to over 34. Across all evaluated languages, both models achieve comparable performance, indicating that the multilingual augmentation strategy successfully transfers the task to Chinese, German, and Italian. While N2+Avg obtains the highest F1 scores on English, German, and Italian, N2+IT+Avg achieves the best MC accuracy, the highest Chinese F1 score, and the strongest BERTScores on German and Italian.

4.6 Re-ranking

To improve generation quality beyond single-pass decoding, we generate $N = 17$ candidate outputs per segment: one greedy decode, one greedy decode with SHAS-based segmentation (Huber et al., 2023; Tsiamas et al., 2022), and 15 sampled candidates. We then apply re-ranking to select the best hypothesis.

A key challenge is that the task identity is unknown at inference time: the re-ranker therefore receives audio, the original prompt instructions, and candidates. This realistic constraint prevents task-specific strategies and requires a single method to generalize across several tasks.

Reranking Strategies. We evaluate six re-ranking strategies: *Likelihood* scores each candidate independently via the model's conditional probability; *Comparison* presents all candidates simultaneously and asks the model to select the best; *Pairwise* runs a sequential tournament where the greedy candidate serves as champion and faces each remaining candidate in turn; *Pairwise Round-Robin* compares every pair of candidates once and selects the candidate with the most wins; *Pairwise Bracket* organises comparisons as a single-elimination bracket, mitigating positional bias from sequential tournaments; and *MBR* selects the candidate with the highest average chrF similarity to all others, requiring no model inference. Additional details, including inference costs, can be found in Tab. A2.

Method	ASR (↓)	SQA (↑)	SSUM (↑)	ST (↑)	Impr.
Greedy	40.77	37.24	29.38	75.28	—
Oracle	-32.10	+14.42	+3.85	+6.11	+14.12
Likelihood	**-24.93**	-11.06	-8.60	+0.93	+1.55
Comparison	+4.16	-4.25	-2.03	-1.35	-2.95
Pairw.	-5.62	-3.78	-1.71	-1.02	-0.22
Pairw. RR	-1.39	-3.36	-1.84	-1.51	-1.33
Pairw. Brack.	-1.83	-3.25	-1.52	-0.63	-0.89
MBR	+3.09	**+0.22**	**-0.79**	+0.27	-0.85
Lik. + MBR	-19.28	-3.33	-2.19	**+1.09**	**+3.71**
Casc. Pairw.	+3.39	-3.32	-2.29	-1.83	-2.71
Casc. Pairw. RR	+14.77	-3.66	-1.45	-0.65	-5.13
Casc. Pairw. Brack.	+12.43	-2.97	-1.76	-1.24	-4.60

Table 3: Re-ranking results. Values show Δ vs. greedy. ASR uses WER (lower ↓ is better); SQA/SSUM/ST use BERTScore/COMET (higher ↑ is better). Casc. = Whisper + Gemma two-stage pipeline; Pairw. = Pairwise; RR = Round-Robin; Lik. + MBR = Likelihood and MBR combined with a tiebreaking pairwise comparison. Greedy row shows absolute scores. Impr. = mean signed improvement (ASR sign-flipped).

Reranking Models. We primarily use `Qwen2.5-Omni`[13] (Xu et al., 2025a) as the re-ranking model, since it can directly condition on the audio input. We additionally experiment with a two-stage pipeline combining `whisper-large-v3`[14] (Radford et al., 2022) for audio transcription and `gemma-3-12b-it`[15] (Team et al., 2025) for text-based candidate comparison, applying all three pairwise strategies.

4.6.1 Reranking Results

Results averaged across languages are shown in Tab. 3. The oracle upper bound reveals substantial potential gains for ASR (−32.1 WER points) and SQA (+14.4), indicating high diversity among the 17 candidates. For SSUM and ST, the oracle improvements are considerably smaller (+3.9 and +2.0 respectively), suggesting less variance and thus limited headroom for re-ranking.

In practice, no single method reliably captures these gains across all tasks. *Likelihood* achieves the strongest ASR improvement (−24.9 WER points), but at the cost of large degradations on SQA and SSUM. *MBR* is conservative: it preserves SQA quality but provides no ASR benefit. Combining both signals via *Lik.+MBR*, which resolves disagreements between Likelihood and MBR through a single pairwise comparison using the same pairwise re-ranking model, retains most of the ASR gain (−19.3) while substantially reducing the SQA degradation (−2.9), yielding the best overall score. Per-language results (Tabs. A3 to A6 Appendix) show consistent gains only for English and Chinese, while German and Italian show no reliable improvement over greedy. For the final submission, we therefore apply *Lik. + MBR* re-ranking only for English and Chinese.

4.6.2 Reranking Analysis

We analyze two aspects of re-ranker behavior that explain the inconsistent results: positional bias and spurious SHAS-candidate selection.

Positional Bias. Pairwise methods are susceptible to spurious preferences for whichever candidate appears first (A-bias) or second (B-bias) in a comparison. We quantify this as

$$P(\text{pick A} \mid \text{B better}) - P(\text{pick B} \mid \text{A better}) \quad (2)$$

where positive values indicate A-bias and negative values B-bias. Results are shown in Tab. A7 in App. C. Sequential *Pairwise* exhibits strong A-bias (+35.3 pp on average, up to +56.8 pp for German), which is expected since the greedy candidate always starts as champion and thus always occupies the first position. *Round-Robin* nearly eliminates this bias (+3.2 pp), since every pair is compared once in each order. *Bracket* introduces a slight B-bias (−5.5 pp) with high variance across languages. These systematic biases likely explain the inconsistent performance of sequential and bracket methods in Tab. 3.

SHAS candidate selection. A second source of error is the spurious selection of the SHAS-segmented candidate for tasks where accurate segmentation provides no benefit. As shown in Tab. A8 in App. C, *Likelihood* selects SHAS frequently for SQA (22.7%) and SSUM (34.5%), directly explaining its large degradations. *MBR* avoids this entirely, the SHAS output tends to be less fluent and therefore scores poorly under chrF-based similarity , but consequently fails to select it for ASR and ST, where it would be beneficial. *Lik.,+,MBR* strikes the best balance: spurious SHAS selection drops sharply for SQA (7.4%) and SSUM (5.5%), while ASR (19.0%) and ST (25.4%) retain it at useful rates. This explains why *Lik.,+,MBR* achieves the best overall score in Tab. 3.

These results suggest a general principle for multi-task re-ranking without task identity: likelihood alone is too aggressive, MBR alone is too conservative, and their combination provides the

[13] Qwen/Qwen2.5-Omni-7B
[14] openai/whisper-large-v3
[15] google/gemma-3-12b-it

Method	EN			ZH		
	SQA↑	SSUM↑	ASR↓	SQA↑	SSUM↑	ST↑
(7)	**44.91**	**23.44**	37.65	39.31	39.57	79.06
(7)+Reranker	44.30	23.10	**21.39**	**40.52**	**40.11**	**80.61**

Table 4: MCIF-long fixed-prompt evaluation comparing N2+IT+Avg and its reranked variant. Best values per column are in **bold**.

Submission	ST COMET	SQA BERT	QE Acc.	ASR WER	SSUM BERT	ACHAP F1
S Primary	**0.844**	**0.484**	0.000	**0.074**	–	–
S Contrastive	0.838	0.448	**0.722**	0.170	–	–
L Primary	0.751	**0.427**	0.000	0.269	**0.275**	**0.474**
L Contrastive	**0.843**	0.344	**0.722**	**0.064**	0.268	0.421

Table 5: Official IWSLT evaluation results averaged across languages. Metrics correspond to COMET (ST), QA-BERTScore (SQA), QE accuracy (QE), WER (ASR), BERTScore (SSUM), and CollarF1 (ACHAP). S for Short Track and L for Long Track.

best tradeoff by using MBR as a regularizer that suppresses spurious likelihood-driven candidate selection. We recommend Lik.+MBR as a default reranking strategy for multi-task speech instruction-following systems where task identity is unavailable at inference time.

4.7 Submission setting

We selected model (7) as our submission setup and additionally apply the re-ranking strategy from Section 4.6. The re-ranking reduced performance on Italian and German; we therefore report only the English and Chinese results on the MCIF long track in Tab. 4. While English shows slight degradation on SQA and SSUM, the ASR improvement is substantial (37.65 → 21.39 WER). Chinese shows consistent improvements across tasks. These results motivate submitting the re-ranking strategy for the English and Chinese tracks.

For the Short Track, we submitted model (7) as the primary system due to its strong WER on pre-segmented audio, with the transcript-based system as the contrastive submission.

4.8 Official IWSLT Evaluation Results

Table 5 summarizes the official IWSLT evaluation results reported by the shared-task organizers (Adelani et al., 2026), averaged across languages. Full language-specific results are provided in Appendix Table A12. The results highlight complementary strengths of the two submissions. In the Long Track, the contrastive system achieves the strongest ST, quality estimation (QE), the surprise task, and ASR performance, while the primary system performs better on SQA, SSUM, and ACHAP. In contrast, the primary submission achieves the strongest Short Track results on ST, SQA, and ASR, whereas the contrastive submission substantially outperforms it on QE. Overall, neither submission consistently outperforms the other across all tasks.

Notably, the primary submissions achieve 0.0 QE accuracy in both tracks, while the contrastive submission reaches 0.722, highlighting a substantial difference between the audio-based and transcript-based pipelines on the unseen QE task.

5 Conclusion

This work presents five contributions to long-form speech instruction following: (1) a general data augmentation framework for constructing long-form training data from short-form sources; (2) an empirical comparison of fixed-probability and temperature-scaled sampling strategies; (3) a negative result on CoT task-token conditioning, revealing a collapse in task discrimination due to weak grounding and task similarity rather than simple data imbalance; (4) a systematic analysis of re-ranking under the realistic constraint of unknown task identity; and (5) a combined Lik.+MBR reranking strategy that mitigates the trade-off between transcription and semantic tasks.

We release our data augmentation code and model checkpoints to support future research on long-form multimodal instruction following [16].

Acknowledgments

This work was supported by the project "How is AI Changing Science? Research in the Era of Learning Algorithms" (HiAICS), funded by the Volkswagen Foundation, and partially by the European Union's Horizon research and innovation programme under grant agreement No. 101135798, project Meetween (My Personal AI Mediator for Virtual MEETtings BetWEEN People) and European Union's Horizon Europe programme grant agreement No. 101213369 (DVPS). The authors gratefully acknowledge computing time provided on HoreKa at the National High-Performance Computing Center at KIT (NHR@KIT), supported by the Federal Ministry of Education and Research, the Ministry of Science, Research and the Arts of Baden-Württemberg, and the DFG.

[16] YapayNet/iwslt2026-if-augmented

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J Hewett, Mojan Javaheripi, Piero Kauffmann, and 1 others. 2024. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*.

David Ifeoluwa Adelani, Antonios Anastasopoulos, Victor Agostinelli, Luisa Bentivogli, Ondřej Bojar, Sebastien Bratières, Marine Carpuat, Roldano Cattoni, Mauro Cettolo, Lizhong Chen, Marcello Federico, Marco Gaido, Mahendra Gupta, HyoJung Han, Ali Hatami, David Javorský, Yejin Jeon, Marek Kasztelnik, Danni Liu, and 26 others. 2026. Speech translation and metrics in 2026: Findings of the iwslt campaign. In *Proceedings of the 23rd International Conference on Spoken Language Translation (IWSLT 2026)*, San Diego, California, US. Association for Computational Linguistics.

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. *Preprint*, arXiv:2402.17733.

Loïc Barrault, Yu-An Chung, Mariano Cora Meglioli, David Dale, Ning Dong, Paul-Ambroise Duquenne, Hady Elsahar, Hongyu Gong, Kevin Heffernan, John Hoffman, and 1 others. 2023. Seamlessm4t: Massively multilingual & multimodal machine translation. *arXiv preprint arXiv:2308.11596*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

Mara Finkelstein, Isaac Caswell, Tobias Domhan, Jan-Thorsten Peter, Juraj Juraska, Parker Riley, Daniel Deutsch, Geza Kovacs, Cole Dilanni, Colin Cherry, and 1 others. 2026. Translategemma technical report. *arXiv preprint arXiv:2601.09012*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3.

Christian Huber, Tu Anh Dinh, Carlos Mullov, Ngoc-Quan Pham, Thai-Binh Nguyen, Fabian Retkowski, Stefan Constantin, Enes Ugan, Danni Liu, Zhaolin Li, and 1 others. 2023. End-to-end evaluation for low-latency simultaneous speech translation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 12–20.

Javier Iranzo-Sánchez, Joan Albert Silvestre-Cerdà, Javier Jorge, Nahuel Roselló, Adrià Giménez, Albert Sanchis, Jorge Civera, and Alfons Juan. 2020. Europarl-st: A multilingual corpus for speech translation of parliamentary debates. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8229–8233.

Paul Kassianik, Baturay Saglam, Alexander Chen, Blaine Nelson, Anu Vellore, Massimo Aufiero, Fraser Burch, Dhruv Kedia, Avi Zohary, Sajana Weerawardhena, and 1 others. 2025. Llama-3.1-foundationai-securityllm-base-8b technical report. *arXiv preprint arXiv:2504.21039*.

Sai Koneru, Fabian Retkowski, Christian Huber, Lukas Hilgert, Seymanur Akti, Enes Yavuz Ugan, Alex Waibel, and Jan Niehues. 2026. Boom: Beyond only one modality kit's multimodal multilingual lecture companion. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 175–187.

Sai Koneru, Maike Züfle, Thai-Binh Nguyen, Seymanur Akti, Jan Niehues, and Alex Waibel. 2025. Kit's offline speech translation and instruction following submission for iwslt 2025. In *Proceedings of the 22nd International Conference on Spoken Language Translation (IWSLT 2025)*, pages 232–244.

Oleksii Kuchaiev, Jason Li, Huyen Nguyen, Oleksii Hrinchuk, Ryan Leary, Boris Ginsburg, Samuel Kriman, Stanislav Beliaev, Vitaly Lavrukhin, Jack Cook, and 1 others. 2019. Nemo: a toolkit for building ai applications using neural modules. *arXiv preprint arXiv:1909.09577*.

Shankar Kumar and William Byrne. 2004. Minimum Bayes-risk decoding for statistical machine translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 169–176, Boston, Massachusetts, USA. Association for Computational Linguistics.

Lujun Li, Lama Sleem, Geoffrey Nichil, Radu State, and 1 others. 2025. Exploring the impact of temperature on large language models: Hot or cold? *Procedia Computer Science*, 264:242–251.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5206–5210. IEEE.

Sara Papi, Maike Züfle, Marco Gaido, Beatrice Savoldi, Danni Liu, Ioannis Douros, Luisa Bentivogli, and Jan Niehues. 2026. MCIF: Multimodal crosslingual instruction-following benchmark from scientific talks. In *The Fourteenth International Conference on Learning Representations*.

Ngoc-Quan Pham, Tuan-Nam Nguyen, Sebastian Stüker, and Alexander Waibel. 2021. Efficient weight

factorization for multilingual speech recognition. *arXiv preprint arXiv:2105.03010*.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust speech recognition via large-scale weak supervision. *Preprint*, arXiv:2212.04356.

Ricardo Rei, Marcos Treviso, Nuno M Guerreiro, Chrysoula Zerva, Ana C Farinha, Christine Maroti, José GC De Souza, Taisiya Glushkova, Duarte Alves, Luisa Coheur, and 1 others. 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.

Fabian Retkowski and Alexander Waibel. 2024. From text segmentation to smart chaptering: A novel benchmark for structuring video transcriptions. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 406–419, St. Julian's, Malta. Association for Computational Linguistics.

Fabian Retkowski, Maike Züfle, Andreas Sudmann, Dinah Pfau, Shinji Watanabe, Jan Niehues, and Alexander Waibel. 2025. Summarizing speech: A comprehensive survey. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 27275–27306, Suzhou, China. Association for Computational Linguistics.

Fabian Retkowski, Maike Züfle, Thai Binh Nguyen, Jan Niehues, and Alexander Waibel. 2026. Beyond transcripts: A renewed perspective on audio chaptering. *Preprint*, arXiv:2602.08979.

Supriti Sinhamahapatra and Jan Niehues. 2025. Do slides help? multi-modal context for automatic transcription of conference talks. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 16111–16121.

Changli Tang, Wenyi Yu, Guangzhi Sun, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun Ma, and Chao Zhang. 2024. SALMONN: towards generic hearing abilities for large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 197 others. 2025. Gemma 3 technical report. *Preprint*, arXiv:2503.19786.

Ioannis Tsiamas, Gerard I. Gállego, José A. R. Fonollosa, and Marta R. Costa-jussà. 2022. SHAS: Approaching optimal Segmentation for End-to-End Speech Translation. In *Proc. Interspeech 2022*, pages 106–110.

Enes Yavuz Ugan, Ngoc-Quan Pham, and Alexander Waibel. 2025. Weight factorization and centralization for continual learning in speech recognition. *arXiv preprint arXiv:2506.16574*.

Changhan Wang, Anne Wu, and Juan Pino. 2020. Covost 2: A massively multilingual speech-to-text translation corpus. *Preprint*, arXiv:2007.10310.

Dingdong Wang, Jincenzi Wu, Junan Li, Dongchao Yang, Xueyuan Chen, Tianhua Zhang, and Helen Meng. 2025. Mmsu: A massive multi-task spoken language understanding and reasoning benchmark. *arXiv preprint arXiv:2506.04779*.

Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, Bin Zhang, Xiong Wang, Yunfei Chu, and Junyang Lin. 2025a. Qwen2.5-omni technical report. *arXiv preprint arXiv:2503.20215*.

Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, Bin Zhang, Xiong Wang, Yunfei Chu, and Junyang Lin. 2025b. Qwen2.5-omni technical report. *Preprint*, arXiv:2503.20215.

Jin Xu, Zhifang Guo, Hangrui Hu, Yunfei Chu, Xiong Wang, Jinzheng He, Yuxuan Wang, Xian Shi, Ting He, Xinfa Zhu, Yuanjun Lv, Yongqi Wang, Dake Guo, He Wang, Linhan Ma, Pei Zhang, Xinyu Zhang, Hongkun Hao, Zishan Guo, and 19 others. 2025c. Qwen3-omni technical report. *Preprint*, arXiv:2509.17765.

Zihan Zhao, Yiyang Jiang, Heyang Liu, Yanfeng Wang, and Yu Wang. 2023. Librisqa: Advancing free-form and open-ended spoken question answering with a novel dataset and framework. *arXiv preprint arXiv:2308.10390*.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 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)*, Bangkok, Thailand. Association for Computational Linguistics.

Maike Züfle and Jan Niehues. 2025. Contrastive learning for task-independent SpeechLLM-pretraining. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 8469–8490, Vienna, Austria. Association for Computational Linguistics.

Maike Züfle, Sara Papi, Beatrice Savoldi, Marco Gaido, Luisa Bentivogli, and Jan Niehues. 2025. NUTSHELL: A dataset for abstract generation from scientific talks. In *Proceedings of the 22nd International Conference on Spoken Language Translation (IWSLT 2025)*, pages 19–32, Vienna, Austria (in-person and online). Association for Computational Linguistics.

A Appendix

B Data

Prompts & Instructions We employ an LLM to generate paraphrased variants of manually designed instruction prompts. For readability, we report a representative subset in Table A1.

System Prompts
(1) You are a helpful assistant specialized in audio and video processing tasks. You follow instructions exactly and don't provide additional explanations or follow-up questions.
(2) You are a helpful assistant specialized in audio processing tasks. You follow instructions exactly and don't provide additional explanations or follow-up questions.

Task	Lang.	#Instr.	Representative instruction variants
ASR	EN	30	(1) Transcribe the entire audio recording into plain text. (2) Generate a faithful transcript of this audio segment. (3) Convert this audio into text exactly as spoken.
ST	DE	10	(1) Konvertieren Sie dieses Audio in deutschen Text. (2) Anhören und ins Deutsche übersetzen. (3) Was wird gesagt? Geben Sie die deutsche Übersetzung an.
	IT	10	(1) Converti questo audio in testo italiano. (2) Ascolta e traduci in italiano. (3) Cosa viene detto? Fornisci la traduzione in italiano.
	ZH	10	(1) 请将其翻译成中文。 (2) 请提供这段录音的中文翻译。 (3) 翻译此音频为中文。
ACHAP	EN	10	(1) Organize this recording into meaningful chapters. For each one, include a Markdown heading (# Title) enclosed by two newlines (\n\n), followed by its transcript. (2)Break this audio into coherent sections as chapters. For every chapter, write a Markdown heading (# Title) with two newlines around it (\n\n), then add the transcript for that part. (3) Create coherent chapter segments for this audio. Each chapter should have a Markdown heading (# Title) with two surrounding newlines (\n\n), then the transcript of that chapter.
	DE	10	(1) Unterteile dieses Audio in zusammenhaengende Kapitel und transkribiere jedes davon. Beginne jedes Kapitel mit einer Markdown-Ueberschrift (# Title), umgeben von zwei Zeilenumbruechen (\n\n), und fuege dann das Transkript an. (2)Segmentiere dieses Audio in zusammenhaengende Kapitel. Fuer jedes Kapitel gib eine Markdown-Ueberschrift (# Title) mit zwei umgebenden Zeilenumbruechen (\n\n) aus und fuege danach das Transkript dieses Abschnitts an. (3) Teile diese Aufnahme in sinnvolle Kapitel auf. Fuer jedes Kapitel schreibe eine Markdown-Ueberschrift (# Title), die von zwei Zeilenumbruechen (\n\n) umgeben ist, und danach den Kapiteltext.
	IT	10	(1) Dividi questo audio in capitoli logici. Ogni capitolo deve iniziare con un'intestazione Markdown (# Title) circondata da due nuove righe (\n\n), seguita dal testo trascritto. (2) Suddividi questa registrazione in capitoli coerenti. Per ogni capitolo, inserisci un'intestazione Markdown (# Title) con due nuove righe attorno (\n\n), poi aggiungi la trascrizione del capitolo. (3) Segmenta questa registrazione seguendo i confini dei capitoli. Per ogni capitolo, usa un'intestazione Markdown (# Title) avvolta da due nuove righe (\n\n), poi fornisci la trascrizione.
	ZH	10	(1) 请将这段音频按逻辑分成多个章节。每章以 Markdown 标题（# Title）开头，标题前后各有两个换行 (\n\n)，并在后面附上对应转写文本。 (2) 请按章节边界对这段录音进行切分。每个章节使用 Markdown 标题（# Title），标题前后各两个换行 (\n\n)，然后提供该章节转写。 (3) 请把这段音频章节化为连贯部分。每个部分都输出 Markdown 标题（# Title），标题前后各两个换行 (\n\n)，然后写该部分转写。
MC	EN	10	(1) Reply with only the choice. (2) Reply with "The answer is" followed by the choice. (3) Reply with "Option" followed by the choice.
	DE	10	(1) Antworte nur mit der Auswahl. (2) Antworte mit „Die Antwort ist" gefolgt von der Auswahl. (3) Antworte mit „Option" gefolgt von der Auswahl.
	IT	10	(1) Rispondi solo con la scelta. (2) Rispondi con "La risposta è" seguito dalla scelta. (3) Rispondi con "Opzione" seguito dalla scelta.
	ZH	10	(1) 只回答选项字母。 (2) 请回答"答案是"加上选项字母。 (3) 请回答"选项"加上选项字母。
SQA	EN	10	(1) Based on the English content, respond to this question with a brief answer: (2) Use the English content to provide a concise answer to the question below: (3) Refer to the English content to answer the question. Be concise:
	DE	20	(1) Beantworte die Frage basierend auf dem Inhalt kurz: (2) Nutze den Inhalt und antworte prägnant auf die folgende Frage: (3) Beantworte die Frage basierend auf dem englischen Inhalt kurz:
	IT	20	(1) In base al contenuto, rispondi brevemente alla domanda: (2) Usa il contenuto per fornire una risposta concisa: (3) In base al contenuto inglese, rispondi brevemente alla domanda:
	ZH	20	(1) 请基于内容，简要回答这个问题： (2) 根据内容，对下列问题给出简洁回答： (3) 请基于英文内容，简要回答这个问题：
SSUM	EN	10	(1) Summarize the English audio. (2) Provide a concise summary of the English audio. (3) Summarize the English audio in at most {x} words.
	DE	10	(1) Fassen Sie das englische Audio zusammen. (2) Erstellen Sie eine knappe Zusammenfassung des englischen Audios. (3) Fassen Sie das englische Audio in maximal {x} Wörtern zusammen.
	IT	10	(1) Riassumi l'audio inglese. (2) Fornisci un riassunto conciso dell'audio in inglese. (3) Riassumi l'audio inglese in massimo {x} parole.
	ZH	10	(1) 概括英文音频内容。 (2) 请简要总结英文音频的主要内容。 (3) 请将英文音频内容概括为不超过{x}个词。

Table A1: Summary of the deduplicated training instruction templates. We report the number of unique templates per task/language and show three representative examples for each setting. NUTSHELL used System Prompt (1); all others used (2). In SSUM, x is replaced by the target summary length (words).

C Reranking Strategies

Setting. We generate $N = 17$ candidates per segment: one greedy decode, one greedy decode with SHAS-based segmentation (Tsiamas et al., 2022), and 15 samples (temperature 0.8). Candidates are produced by our fine-tuned model, while re-ranking is performed by the base `Qwen2.5-Omni-7B` (Xu et al., 2025a), which can directly condition on the audio input. For the cascaded methods, audio is first transcribed with `whisper-large-v3` (Radford et al., 2022) and the resulting transcript is passed together with all candidates to `gemma-3-12b-it` (Team et al., 2025) for text-based comparison. An overview of the re-ranking strategies and the number of model calls required for each is given in Tab. A2.

Method	Description	Model calls
Likelihood	Each candidate is scored independently by computing $\log p(\text{candidate} \mid \text{audio})$ with an external model. The candidate with the highest likelihood is selected.	N
Comparison	The model receives the audio, the task, and all candidates simultaneously and is prompted to identify the best candidate.	1
Pairwise	A sequential tournament: candidate 0 (greedy) acts as champion and is compared against candidates 1 through $N-1$ in order; the winner of each comparison advances.	$N-1$
Pairwise Round-Robin	Every pair of candidates is compared once; the candidate with the most wins is selected. Unbiased with respect to candidate order.	$\frac{N(N-1)}{2}$
Pairwise Bracket	A single-elimination bracket where winners are matched against winners across $\log_2 N$ rounds, avoiding the positional bias of the sequential tournament.	$N-1$
MBR	Minimum Bayes Risk (Kumar and Byrne, 2004): the candidate with the highest average chrF similarity to all other candidates is selected.	0

Table A2: Overview of re-ranking methods. For model-based methods, the original instruction is included in the prompt of the reranker since the task is determined at inference time.

Per-language results. We use the MCIF test set (Papi et al., 2026) to evaluate our re-ranking methods using the official MCIF evaluation code. Results per language are shown in Tab. A3 (en), Tab. A4 (de), Tab. A5 (it), and Tab. A6 (zh). Re-ranking yields consistent improvements only for English, where Likelihood achieves the largest gain, driven primarily by a strong improvement on ASR. For Chinese, modest improvements are observed with Likelihood and MBR. For German and Italian, no method reliably improves over greedy across all tasks.

Analysis. Tab. A7 reports positional bias for each pairwise re-ranking method and language, measured as

$$P(\text{pick A} \mid \text{B better}) - P(\text{pick B} \mid \text{A better}). \tag{3}$$

Positive values indicate a spurious preference for the first candidate, negative values for the second. Tab. A8 reports the proportion of segments for which each re-ranking method selects the SHAS-segmented candidate, broken down by task. Both tables complement the analysis in Section 4.6.

D Speech Translation

We perform English to German, Italian, and Chinese speech translation using `translategemma-12b-it` (Finkelstein et al., 2026). Input speech is first transcribed into English text and embedded within a structured conversational format, from which the relevant response segments are extracted for translation. To efficiently handle large-scale datasets, we employ a streaming JSON parser (ijson), enabling memory-efficient sequential processing.

Inference is conducted in batches of size 8, with the model loaded in bfloat16 precision. Inputs are formatted using the model-specific chat template and tokenized with dynamic padding before being transferred to the appropriate device. Decoding is performed using deterministic (greedy) generation with a maximum output length depending on the dataset. To improve generation stability and mitigate repetition artifacts, we apply a repetition penalty around 1.1 and enforce a 5-gram repetition constraint.

Method	ASR ↓	SQA ↑	SSUM ↑	Impr.
Greedy	40.77	38.43	23.43	—
Oracle	-32.10	+14.84	+5.51	+17.48
Likelihood	**-24.93**	-9.21	-2.52	+4.40
Comparison	+4.16	-4.63	-2.10	-3.63
Pairw.	-5.62	-3.19	-1.94	+0.16
Pairw. RR	-1.39	-3.37	-2.07	-1.35
Pairw. Brack.	-1.83	-2.67	-1.32	-0.72
MBR	+3.09	**+1.59**	**-0.41**	-0.64
Lik. + MBR	-19.28	-2.94	-2.18	**+4.72**
Casc. Pairw.	+3.39	-2.46	-2.28	-2.71
Casc. Pairw. RR	+14.77	-3.74	-1.50	-6.67
Casc. Pairw. Brack.	+12.43	-1.98	-1.46	-5.29

Table A3: Re-ranking results — English. Values show Δ vs. greedy. ASR uses WER (lower ↓ is better); QA/SUM/ST use BERTScore/COMET (higher ↑ is better). Casc. = Whisper + Gemma two-stage pipeline; Pairw. = Pairwise; RR = Round-Robin; Lik. + MBR = Likelihood and MBR combined with a tiebreaking pairwise comparison. Greedy row shows absolute scores. Impr. = mean signed improvement (ASR sign-flipped). **Bold**: best method per column.

Method	SQA ↑	SSUM ↑	ST ↑	Impr.
Greedy	36.77	25.60	72.91	—
Oracle	+15.76	+4.06	+5.49	+8.44
Likelihood	-14.29	-8.01	**+2.77**	-6.51
Comparison	-2.55	-2.55	-1.47	-2.19
Pairw.	-2.68	-1.09	-1.22	-1.66
Pairw. RR	-2.26	-1.22	-1.05	-1.51
Pairw. Brack.	-1.79	**-0.86**	-1.25	-1.30
MBR	**+0.70**	-1.03	-0.46	**-0.26**
Casc. Pairw.	-1.98	-2.05	-1.46	-1.83
Casc. Pairw. RR	-2.98	-1.42	-0.91	-1.77
Casc. Pairw. Brack.	-1.69	-1.74	-1.51	-1.65

Table A4: Re-ranking results — German. Values show Δ vs. greedy. ASR uses WER (lower ↓ is better); QA/SUM/ST use BERTScore/COMET (higher ↑ is better). Casc. = Whisper + Gemma two-stage pipeline; Pairw. = Pairwise; RR = Round-Robin. Greedy row shows absolute scores. Impr. = mean signed improvement (ASR sign-flipped). **Bold**: best method per column.

Method	SQA ↑	SSUM ↑	ST ↑	Impr.
Greedy	35.60	28.38	74.80	—
Oracle	+13.76	+2.72	+6.27	+7.58
Likelihood	-16.13	-8.78	**+1.19**	-7.91
Comparison	-5.05	-2.26	-3.06	-3.46
Pairw.	-4.32	-2.53	-2.22	-3.02
Pairw. RR	-4.46	-2.23	-1.96	-2.88
Pairw. Brack.	-4.99	-2.66	-1.72	-3.12
MBR	**-1.49**	**-1.74**	-0.33	**-1.19**
Lik. + MBR	-5.03	-3.21	+0.43	-2.60
Casc. Pairw.	-2.80	-3.18	-4.70	-3.56
Casc. Pairw. RR	-3.22	-2.39	-1.80	-2.47
Casc. Pairw. Brack.	-2.51	-2.52	-3.56	-2.86

Table A5: Re-ranking results — Italian. Values show Δ vs. greedy. ASR uses WER (lower ↓ is better); QA/SUM/ST use BERTScore/COMET (higher ↑ is better). Casc. = Whisper + Gemma two-stage pipeline; Pairw. = Pairwise; RR = Round-Robin; Lik. + MBR = Likelihood and MBR combined with a tiebreaking pairwise comparison. Greedy row shows absolute scores. Impr. = mean signed improvement (ASR sign-flipped). **Bold**: best method per column.

Method	SQA ↑	SSUM ↑	ST ↑	Impr.
Greedy	38.17	40.09	78.12	—
Oracle	+13.32	+3.11	+6.56	+7.66
Likelihood	-4.61	-15.10	-1.16	-6.96
Comparison	-4.78	-1.20	+0.48	-1.83
Pairw.	-4.94	-1.28	+0.38	-1.95
Pairw. RR	—	—	—	—
Pairw. Brack.	-3.55	-1.24	+1.09	-1.23
MBR	**+0.09**	**+0.01**	+1.61	**+0.57**
Lik. + MBR	+0.04	-0.43	**+1.65**	+0.42
Casc. Pairw.	-6.03	-1.65	+0.68	-2.33
Casc. Pairw. RR	-4.68	-0.47	+0.77	-1.46
Casc. Pairw. Brack.	-5.69	-1.31	+1.35	-1.88

Table A6: Re-ranking results — Chinese. Values show Δ vs. greedy. ASR uses WER (lower $\downarrow$ is better); QA/SUM/ST use BERTScore/COMET (higher $\uparrow$ is better). Casc. = Whisper + Gemma two-stage pipeline; Pairw. = Pairwise; RR = Round-Robin; Lik. + MBR = Likelihood and MBR combined with a tiebreaking pairwise comparison. Greedy row shows absolute scores. Impr. = mean signed improvement (ASR sign-flipped). **Bold**: best method per column.

Method	en	de	it	zh	Avg. Bias
Comparison	8.5 / 16 (last)	7.4 / 16 (first)	7.7 / 16 (first)	7.6 / 16 (first)	7.8 / 16 (first)
Pairw.	+30.0 (A)	+56.8 (A)	+49.3 (A)	+5.1 (A)	+35.3 (A)
Pairw. RR	+2.3 (A)	+4.0 (A)	+3.4 (A)	—	+3.2 (A)
Pairw. Brack.	−8.6 (B)	+11.3 (A)	+1.9 (A)	−26.8 (B)	−5.5 (B)
Casc. Pairw.	+19.0 (A)	+20.8 (A)	+17.8 (A)	+10.6 (A)	+17.1 (A)
Casc. Pairw. RR	+1.1 (A)	+3.6 (A)	+2.5 (A)	+2.1 (A)	+2.3 (A)
Casc. Pairw. Brack.	−21.2 (B)	−21.5 (B)	−24.9 (B)	−29.1 (B)	−24.2 (B)

Table A7: Positional preference of re-ranking methods. For pairwise methods: position bias $= P(\text{pick A} \mid \text{B oracle better}) - P(\text{pick B} \mid \text{A oracle better})$, with preferred position in parentheses (A = first, B = second); values near 0 indicate no bias. For *Comparison*: average selected candidate index (0 = greedy, 16 = SHAS), where 8 would be unbiased. Casc. = Whisper + Gemma cascade; Pairw. = Pairwise; RR = Round-Robin.

Method	ASR	SQA	SSUM	ST
Likelihood	19.0	22.7	34.5	36.5
MBR	0.0	0.0	0.0	0.0
Lik. + MBR	19.0	7.4	5.5	25.4

Table A8: Rate at which the SHAS-segmented candidate is selected per task (%). Ideally the re-ranker should prefer SHAS for ASR and ST (where accurate segmentation helps) but not for QA and SUM.

E Results

Per language evaluatoins While our main results (Section 4.5) present language-averaged scores, Tables A9 and A10 provide per-language breakdowns for all four languages on the MCIF long-form track with fixed and mixed prompts respectively. Table A11 shows MC results (English only) and ACHAP results (all languages). These analyses reveal substantial performance variation across languages, with some configurations (e.g., row 7) generalizing robustly while others (e.g., row 5) excel on specific languages. This granular view helps identify how language-specific challenges—morphological complexity, data scarcity, and pre-training bias—affect different tasks and languages.

		EN			ZH			DE			IT		
ID	**Method**	SQA	SSUM	ASR	SQA	SSUM	ST	SQA	SSUM	ST	SQA	SSUM	ST
Baselines													
(1)	Omni	24.93	19.34	53.40	43.03	9.59	75.88	28.34	11.55	61.16	26.82	16.35	68.92
(2)	Cascaded	25.07	19.36	5.88	26.68	10.31	84.66	31.64	11.15	76.55	26.53	12.41	81.23
Fine-tuned													
(3)	Temp	37.75	23.03	30.59	36.83	39.43	78.24	35.47	24.86	73.42	34.58	27.75	75.08
(4)	T=2	41.46	22.10	27.58	38.77	39.64	80.20	35.22	25.58	73.35	35.53	28.14	74.75
(5)	N2	44.41	23.95	25.98	41.51	39.44	78.20	37.63	25.47	72.33	36.36	27.32	74.40
(6)	N2+Avg	45.00	23.85	29.36	39.82	26.23	72.42	39.82	26.23	72.42	38.09	27.93	75.21
(7)	N2+IT+Avg	44.91	23.44	37.65	39.64	26.30	73.10	39.64	26.30	73.10	37.49	28.36	75.62
(8)	N2+IT+Avg+SHAS	-19.48	0.87	11.72	2.63	-2.78	84.32	-10.50	5.25	77.83	-16.99	8.38	79.72
(9)	CoT	34.49	24.29	79.24	35.06	26.70	39.97	35.06	26.70	39.97	34.92	28.53	46.33
(10)	Cascaded+FT	33.02	22.02	5.90	38.50	39.82	85.00	31.57	23.74	81.75	30.36	28.62	84.42

Legend: (3) temperature-based sampling; (4) temperature-based sampling with $T = 2$; (5) continued fine-tuning on in-domain NUTSHELL data after (4); (6) checkpoint averaging of (5); (7) continued fine-tuning on Italian NUTSHELL data with checkpoint averaging; (8) (7) with SHAS; (9) chain-of-thought prompting; (10) cascaded system with fine-tuning.

Table A9: MCIF-long fixed-prompt evaluation results across tasks (SQA, SSUM, ASR, ST) and languages.

		EN			ZH			DE			IT		
ID	**Method**	SQA	SSUM	ASR	SQA	SSUM	ST	SQA	SSUM	ST	SQA	SSUM	ST
Baselines													
(1)	Omni	33.15	17.57	35.35	39.59	21.25	75.74	30.37	16.04	66.61	28.66	16.63	70.01
(2)	Cascaded	26.97	18.12	6.85	28.87	12.51	84.40	28.06	11.08	76.70	26.72	11.88	80.28
Fine-tuned													
(3)	Temp	39.37	22.67	33.61	36.60	39.81	78.72	35.35	24.75	72.84	34.34	27.79	73.15
(4)	T=2	42.30	22.18	38.43	38.38	39.51	79.60	34.87	26.16	73.45	35.92	28.10	74.82
(5)	N2	44.46	23.35	30.46	41.34	39.26	78.35	38.20	25.12	72.92	35.93	27.25	73.56
(6)	N2+Avg	45.15	24.27	34.48	40.80	39.37	79.60	40.25	26.07	72.71	37.23	27.62	73.96
(7)	N2+IT+Avg	45.15	23.75	36.88	40.70	39.50	79.11	38.33	26.13	73.39	37.04	28.25	75.54
(8)	N2+IT+SHAS	-17.74	0.74	11.25	-0.95	7.48	84.25	-13.47	6.89	77.98	-16.39	8.47	80.30
(9)	CoT	37.29	24.06	80.04	35.30	39.88	44.82	34.74	26.56	41.03	34.66	28.44	45.22
(10)	Cascaded FT	32.49	22.06	9.76	37.62	38.03	85.19	29.40	21.68	81.58	28.83	27.56	84.47

Legend: (3) temperature-based sampling; (4) temperature-based sampling with $T = 2$; (5) continued fine-tuning on in-domain NUTSHELL data after (4); (6) checkpoint averaging of (5); (7) continued fine-tuning on Italian NUTSHELL data with checkpoint averaging; (8) (7) with SHAS; (9) chain-of-thought prompting; (10) cascaded system with fine-tuning.

Table A10: MCIF-long mixed-prompt evaluation results across tasks (SQA, SSUM, ASR, ST) and languages.

Method	**MC (EN)↑**	**EN (ACHAP)**		**ZH**		**DE**		**IT**	
	Acc	F1	BERT	F1	BERT	F1	BERT	F1	BERT
Baseline	0.00	1.20	86.60	–	–	–	–	–	–
(6) N2+Avg	81.53	**35.86**	**87.84**	10.11	**62.71**	**45.46**	69.58	**40.21**	69.37
(7) N2+IT+Avg	**81.83**	34.39	86.75	**10.63**	62.15	44.94	**69.95**	39.11	**69.84**

Table A11: MC accuracy (EN only) and ACHAP performance across languages. For ACHAP, we report F1 and BERTScore (GC). The baseline is only evaluated on English due to poor performance on other languages. Best values per column are in **bold**.

			ST	SQA	QE		ASR	SSUM	ACHAP			
Track	**Submission**	**Lang**	COMET	BERT	Acc.	Fmt.	WER	BERT	WER	COMET	F1	BERT
Short Track												
Short	Primary	EN	–	0.495	–	–	0.074	–	–	–	–	–
Short	Primary	ZH	0.852	0.456	0.000	0.000	–	–	–	–	–	–
Short	Primary	DE	0.840	0.466	0.000	0.000	–	–	–	–	–	–
Short	Primary	IT	0.841	0.519	–	–	–	–	–	–	–	–
Short	Contrastive	EN	–	0.450	–	–	0.170	–	–	–	–	–
Short	Contrastive	ZH	0.845	0.471	0.739	0.993	–	–	–	–	–	–
Short	Contrastive	DE	0.830	0.423	0.705	1.000	–	–	–	–	–	–
Short	Contrastive	IT	0.830	0.449	–	–	–	–	–	–	–	–
Long Track												
Long	Primary	EN	–	0.412	–	–	0.269	0.212	0.311	–	0.436	0.869
Long	Primary	ZH	0.789	0.452	0.000	0.000	–	0.383	–	0.785	0.503	0.685
Long	Primary	DE	0.733	0.405	0.000	0.000	–	0.238	–	0.740	0.500	0.690
Long	Primary	IT	0.732	0.439	–	–	–	0.267	–	0.737	0.456	0.703
Long	Contrastive	EN	–	0.385	–	–	0.064	0.218	0.093	–	0.583	0.877
Long	Contrastive	ZH	0.847	0.360	0.739	0.993	–	0.378	–	0.451	0.103	0.511
Long	Contrastive	DE	0.840	0.308	0.705	1.000	–	0.208	–	0.836	0.508	0.676
Long	Contrastive	IT	0.841	0.322	–	–	–	0.269	–	0.842	0.489	0.709

Table A12: Official IWSLT 2026 evaluation results for our submitted systems. ST is evaluated using COMET, SQA using QA-BERTScore, QE using accuracy and format accuracy, ASR using WER, and SSUM using BERTScore. For ACHAP, we report WER, COMET, CollarF1, and GC-BERTScore.