AlignAtt4LLM: Fast AlignAtt for Decoder-Only LLMs at IWSLT 2026 Simultaneous Speech Translation Task

Quentin Fuxa
Independent Researcher
quentin.fuxa@gmail.com

Dominik Macháček
Charles University, MFF, ÚFAL
& University of Edinburgh
machacek@ufal.mff.cuni.cz

Abstract

We describe AlignAtt4LLM, an IWSLT 2026 simultaneous speech translation system for English to German, Italian, and Chinese. The system is a synchronous cascade: Qwen3-ASR with forced alignment produces an incrementally updated source transcript, and Gemma-4 E4B-it translates that prefix under an MT-side AlignAtt policy.

To our knowledge, this is the first application of AlignAtt to a decoder-only LLM, where the encoder-decoder cross-attention used by earlier AlignAtt systems is absent. We recover a usable policy by proposing (1) an explicit source span in the prompt, (2) offline selection of translation-specific alignment heads, (3) selective *qk-fast* replay of the draft-to-source attention block, and (4) runtime query/key capture that preserves model outputs bit-identically.

On the IWSLT 2026 development set, AlignAtt4LLM outperforms the supplied baselines for the European target languages, English to German and English to Italian, in both the low-latency regime around 2 seconds and the high-latency regime below 4 seconds CU-LongYAAL. Results for English to Chinese are more mixed, but the method is not tied to Gemma-4: because AlignAtt4LLM only requires a deterministic prompt layout, calibrated attention heads, and query/key capture, the same policy can be reapplied to stronger translation-focused decoder-only MT backbones for non-European target languages.

1 Introduction

This paper describes the AlignAtt4LLM IWSLT 2026 submission for English to German, Italian, and Chinese simultaneous speech translation (Adelani et al., 2026). The system is a synchronous cascade: Qwen3-ASR with the Qwen3 forced aligner (Shi et al., 2026) produces an incrementally updated source transcript with word end times, and Gemma-4 E4B-it (Google DeepMind, 2026) translates the current source text under an MT-side AlignAtt policy.

The central contribution is to make an offline decoder-only LLM usable in simultaneous mode. Standard AlignAtt reads encoder-decoder cross-attention, but decoder-only LLMs have no such cross-attention. We instead expose the source transcript as a known prompt span, select a small set of translation-specific self-attention heads offline, and accept only draft prefixes whose reconstructed attention signal remains within the currently available source frontier.

The second contribution is making the implementation fast enough for computational-aware (CA) evaluation. Both ASR and MT are served through vLLM (Kwon et al., 2023), which gives the cascade one high-throughput inference stack and hot model reuse across chunks. On the MT side, this also means the policy cannot inspect a Python-visible attention matrix: self-attention is hidden inside fused kernels. We therefore capture the exact query/key tensors consumed by the deployed Gemma attention module and replay only the draft-to-source block needed by AlignAtt.

Our implementation is available at `https://github.com/QuentinFuxa/AlignAtt4LLM`.

Section 2 describes the context of the task and prior work, Section 3 describes the cascade, Section 4 details the MT-side AlignAtt realization, and Section 5 reports the evaluation.

2 Background

AlignAtt (Papi et al., 2023, 2024) is a simultaneous policy that lets an offline sequence-to-sequence model operate in simultaneous mode. At each generation step, the policy derives a source position for the next target token from attention and stops generation when that position crosses the accessible-source frontier. Earlier AlignAtt systems obtained

Proceedings of the 23rd International Conference on Spoken Language Translation (IWSLT 2026), pages 284–295

this signal from encoder-decoder cross-attention. AlignAtt is currently recognized as state of the art, it has been used in e.g. by the top-performing IWSLT 2025 system (Macháček and Polák, 2025).

Offline models in simultaneous mode. Repurposing offline models for simultaneous translation is a highly promising approach because it allows reusing high-quality, multilingual, instruction-following models without training dedicated simultaneous models for every language direction. However, the disadvantage is that an offline model is not trained to decide when a partial source prefix is sufficient, and naive re-translation can flicker or hallucinate on incomplete input. Simultaneous policies such as AlignAtt provide the missing commit decision while preserving the quality and flexibility of the offline model (Papi et al., 2023; Macháček and Polák, 2025).

Decoder-only LLMs have recently become central in high-quality text-to-text MT systems (Kocmi et al., 2024, 2025). Prior simultaneous systems have already used instruction-tuned decoder-only LLMs by prompting them with the translation direction, current source prefix, and already emitted target prefix (Macháček and Polák, 2025). Long context, instruction following, and in-context examples make this family attractive for robust MT, but previous decoder-only simultaneous implementations used LocalAgreement (Polák et al., 2022, 2023) rather than AlignAtt. LocalAgreement is a strong simultaneous policy, but it does not take advantage of model-internal attention and generally introduces more latency than AlignAtt (Macháček and Polák, 2025).

Translation-specific attention heads. Liu et al. (2026) show that multilingual decoder-only LLMs contain a sparse set of heads whose attention argmaxes track source-target word alignments. This suggests a decoder-only AlignAtt path, but only if two runtime conditions are met: the source span must be identifiable in the prompt, and the policy-visible attention rows must be reconstructed from the same tensors used by the deployed inference engine. Our proposed method supplies both conditions.

2.1 IWSLT 2026 Simultaneous Task

Simulation environment. The IWSLT 2026 Simultaneous Task proposes the Simulstream toolkit (Gaido et al., 2025) for simulating simultaneous translation sessions. We follow the long-form, unsegmented task setup, which allows re-translation at the evaluation interface, but AlignAtt4LLM itself emits append-only incremental output, the mode preferred in real-world deployments where re-translation flicker could disrupt readers following the live text.

Development data. We use the official IWSLT 2026 MCIF development set. We report translation quality with BLEU (Papineni et al., 2002), chrF (Popović, 2015), and XCOMET-XL (Guerreiro et al., 2024). Translation latency is reported with LongYAAL (Polák et al., 2025) in both computational-unaware (CU) and computational-aware (CA) variants.

3 AlignAtt4LLM System Overview

ASR. The speech component of the cascade uses two upstream Qwen components: Qwen3-ASR-1.7B transcribes the live audio tail, and Qwen3-ForcedAligner-0.6B assigns word-level start and end times to that transcript (Shi et al., 2026). At every chunk boundary, the cascade re-transcribes the current live utterance tail and invokes the forced aligner with timestamp output enabled; the word times are therefore produced online inside our simulation run, not added later as an offline post-processing step. Adjacent ASR hypotheses are stabilized by a longest-common-prefix commit rule up to sentence-final punctuation, while the remaining live tail is allowed to change on the next chunk. We retain this dedicated aligner because it performed better than all the alternative candidates summarized in Appendix A.

MT. The translation component uses Gemma-4 E4B-it (Google DeepMind, 2026) served through vLLM (Kwon et al., 2023). With every source update, Gemma receives the current transcript prefix, the already accepted target prefix, and a fixed translation instruction, then greedily generates a draft continuation of at most 16 new tokens. AlignAtt accepts the longest draft prefix whose alignment signal stays on the currently accessible side of the source frontier.

Synchronization. The cascade is synchronous and chunk-based. Each chunk first updates the ASR transcript prefix and then launches one MT request. Source words are considered accessible only once their end time is older than a hold-back margin; in the official IWSLT runs, this margin is 0 ms, so a word becomes accessible as soon as its aligned end time is observed (see Section 5.3

for a reliability analysis suggesting a conservative 250 ms tail hold-back). Both regimes also delay the first MT emission until 2 seconds of source audio have accumulated.

This conservative synchronous regime cleanly separates policy behavior from scheduler overlap and defines the CA setting used in our experiments. Figure 1 summarizes this chunk-synchronous runtime path. Appendix B show alternative synchronization modes.

Latency-quality parameters. We keep two official latency regimes. The low-latency submission uses $\Delta_{\text{chunk}} = 850$ ms and the high-latency submission uses $\Delta_{\text{chunk}} = 1500$ ms. Both use the same ASR backend, Gemma-4 MT backbone, and MT-side AlignAtt realization. They also use the same per-direction top-8 MT head sets, a width-7 source-axis median filter applied before taking the source-side argmax, and a one-source-token border margin $b = 1$.

4 AlignAtt for Decoder-Only LLMs

This section focuses on the method that enables AlignAtt for decoder-only LLMs.

4.1 Prompt Layout

Standard AlignAtt does not directly apply because a decoder-only LLM has no decoder cross-attention. We therefore make the source span explicit in the causal prompt. At chunk k, the serialized chat prompt is a concatenation of the system prompt, live transcript prefix, translation instruction, accepted translation prefix, and current draft:

$$\mathbf{p}^{(k)} = \left[\mathbf{p}^{\text{sys}} \| \mathbf{s}^{(k)} \| \mathbf{p}^{\text{instr}} \| \mathbf{y}_{1:m_k} \| \widetilde{\mathbf{y}}^{(k)}\right], \quad (1)$$

where $\mathbf{s}^{(k)}$ is the live transcript prefix returned by ASR, $\mathbf{y}_{1:m_k}$ is the already accepted translation prefix, and $\widetilde{\mathbf{y}}^{(k)}$ is the draft proposed by Gemma-4 at the current step. The transcript prefix is therefore a contiguous prompt span with a known map $\phi^{(k)}$ from source-word indices to prompt positions.

Given this prompt layout, Figure 2 summarizes the conceptual shift: the policy intuition remains the same as in encoder-decoder AlignAtt, but the alignment signal must now be recovered from a prompt-structured self-attention substrate rather than read directly from cross-attention.

4.2 Selection of Alignment Heads

Not all self-attention heads carry a useful translation signal. Following Liu et al. (2026), we operationalize this with a two-stage offline calibration procedure under the exact prompt layout of Eq. (1): GPT-5-mini (OpenAI, 2025) first provides word-level source-target alignments on held-out parallel text, then every Gemma head is scored with Translation Score (TS), the aligned-token argmax accuracy used by Liu et al. (2026), on those aligned examples. We retain the top $k = 8$ heads per language pair. This head set is fixed at inference time and is the only part of the policy that depends on offline calibration. The retained MT heads are sparse and concentrated in a limited late-depth region of the backbone rather than being spread uniformly; Appendix C visualizes that pattern.

The policy only reads the source slice of each draft attention row, $\mathbf{A}^{(\ell,h)}_{t,\phi^{(k)}(s)}$, but unlike encoder-decoder cross-attention, this slice is not source-normalized. Off-source mass on the accepted target prefix, prompt template, and speculative suffix is therefore structural rather than implementation noise. On the 78 text-only qualitative probes used for diagnostics, drafted units allocate on average 9% to accessible source tokens, 8% to inaccessible source tokens, 81% to non-source prompt positions, and 2% to the speculative suffix; among units that the policy actually accepts, the inaccessible-source share drops from 8% to 1%.

4.3 Selective Reconstruction for the Policy

The policy never needs the full $n \times n$ self-attention matrix. It only needs draft rows against source columns. For selected heads (ℓ, h), draft positions t, and source positions s, we reconstruct

$$\widehat{A}^{(\ell,h)}_{t,s} = \frac{\exp\left(\gamma^{(\ell)} q^{(\ell,h)}_t \cdot k^{(\ell,h)}_{\phi^{(k)}(s)} + m_{t,\phi^{(k)}(s)}\right)}{\sum_j \exp\left(\gamma^{(\ell)} q^{(\ell,h)}_t \cdot k^{(\ell,h)}_j + m_{t,j}\right)}, \quad (2)$$

where q and k are the captured post-normalization tensors after rotary position embedding (RoPE; Su et al., 2021) actually consumed by the deployed attention module, $\gamma^{(\ell)}$ is that module's runtime scaling factor, and $m_{t,j}$ replays the same causal and sliding-window masks as the fused forward. We call this replay path *qk-fast*: it reconstructs only the policy-visible block and matches the deployed attention algebra up to floating-point reassociation relative to the fused forward. Figure 3 illustrates the capture-and-replay path: the fused runtime keeps the full attention matrix internal, while the observer stores only the selected draft queries and prompt keys needed to recompute the source-facing policy

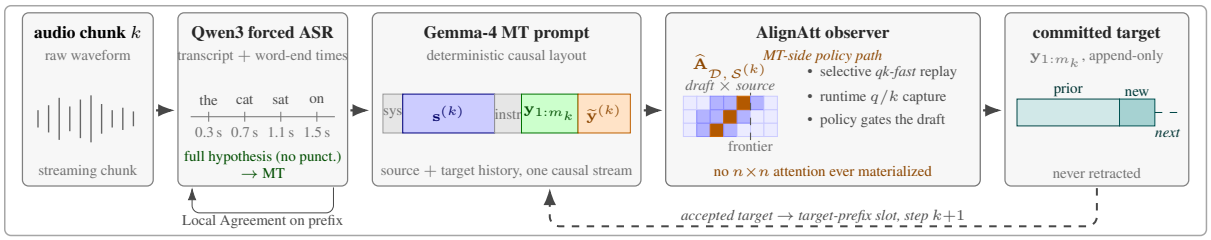

Figure 1: **Chunk-synchronous cascade, one step.** Each chunk first updates the source prefix with Qwen3 forced ASR, then runs one Gemma-4 MT step. The MT prompt keeps the ASR transcript explicit inside the decoder-only causal layout, so the observer can capture the selected heads' queries and keys on the deployed vLLM path and reconstruct only the draft-to-source block $\widehat{\mathbf{A}}_{\mathcal{D},\mathcal{S}^{(k)}}$ used by the acceptance policy. Accepted target tokens are appended to $\mathbf{y}_{1:m_k}$ and become the target-prefix slot of the next step's prompt (dashed teal feedback).

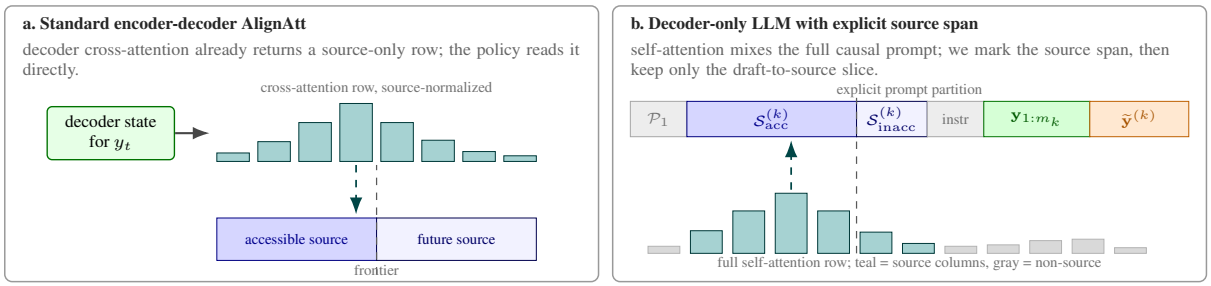

Figure 2: **How AlignAtt changes substrate between encoder-decoder and decoder-only models.** *(a)* In the original encoder-decoder setting, the decoder already exposes a source-only cross-attention row, so the policy can gate tokens directly against the accessible source frontier: if the peak of the row falls on the accessible side, the draft token is accepted. *(b)* In our decoder-only MT setting, source and target history share one causal prompt. We therefore mark the source span explicitly in prompt space, isolate translation-specific heads, and reconstruct only the draft-to-source slice needed by the policy; the decision rule is otherwise the same as in (a).

block.

From these per-head rows we build two source-side provenance features for each drafted token: accessible-source mass π_t^{acc} and inaccessible-source mass π_t^{inacc},

$$\pi_t^{\text{acc}} = \sum_{s \in \mathcal{S}^{(k)}_{\text{acc}}} \bar{\mathbf{r}}_t(s), \tag{3}$$

$$\pi_t^{\text{inacc}} = \sum_{s \in \mathcal{S}^{(k)}_{\text{inacc}}} \bar{\mathbf{r}}_t(s), \tag{4}$$

where $\bar{\mathbf{r}}_t$ is the head-averaged reconstructed row. We additionally normalize per-head rows online with prefix Welford statistics and apply a short median filter along the source axis before taking the source-side argmax; these operations stabilize the alignment peak without changing the underlying replayed rows. The runtime gate only thresholds π_t^{acc}; the complementary π_t^{inacc} is retained as a diagnostic split in Appendix D.

4.4 AlignAtt Acceptance Policy

The MT policy is a first-failure scan over drafted tokens. Let $\hat{s}_t$ be the source-side argmax of the filtered row for draft token t, and let $N^{(k)}_{\text{acc}}$ be the number of source words whose ASR end time is already on the accessible side of the frontier. The scan stops when one of three conditions first fires:

$$\text{SOURCE-FRONTIER}: \quad \hat{s}_t \geq N^{(k)}_{\text{acc}} + b, \tag{5}$$

$$\text{ARGMAX-MASS-WEAK}: \quad \bar{\mathbf{r}}_t(\hat{s}_t) < \tau_{\text{argmax}}, \tag{6}$$

$$\text{PROVENANCE-WEAK}: \quad \pi_t^{\text{acc}} < \tau_{\text{src}}. \tag{7}$$

In the official 850/1500 ms operating points, $b = 1$, $\tau_{\text{argmax}} = 0$, and $\tau_{\text{src}} = 0$, so the optional argmax-mass and minimum-source-mass gates are present in the runtime but inactive in these operating points.

Figure 4 makes the decision path explicit. One branch averages the retained heads and sums accessible-source mass to obtain π_t^{acc}. The other normalizes and smooths the same replayed rows before taking the source-side argmax $\hat{s}_t$. The gate accepts token t if the three tests in Eqs. (5) to (7) all pass; a left-to-right first-failure scan then emits the longest accepting draft prefix and rounds back to the last completed stability unit. Here a stability unit is the smallest target fragment that the

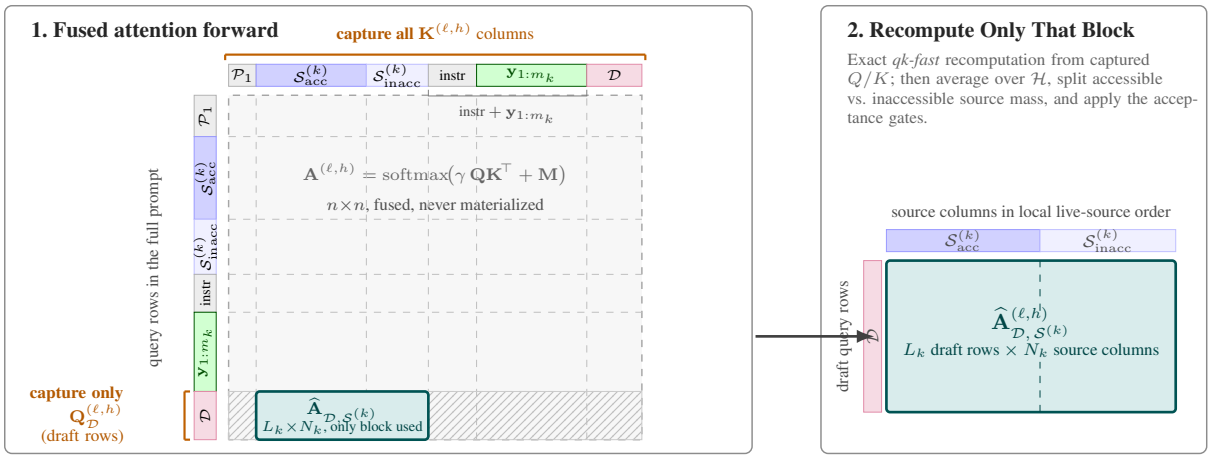

Figure 3: **Selective reconstruction with runtime capture.** The $n \times n$ attention matrix $\mathbf{A}^{(\ell,h)}$ is executed entirely inside the fused attention kernel, so the full $n \times n$ matrix is never materialized. For each AlignAtt head $(\ell, h) \in \mathcal{H}$, the observer copies $\mathbf{K}^{(\ell,h)}$ for all positions and $\mathbf{Q}^{(\ell,h)}$ only for the draft rows $\mathcal{D}$ into fixed-shape buffers. The green band $\mathbf{y}_{1:m_k}$ marks the committed target prefix that grows across chunks; hatched cells are never reconstructed. After the forward, Eq. (2) recomputes exactly only the draft-to-source block $\widehat{\mathbf{A}}_{\mathcal{D},\mathcal{S}^{(k)}}$, which is the sole object consumed by the source-side provenance split of Eqs. (3) to (4), where π_t^{acc} drives the gate and π_t^{inacc} is retained for diagnostics, and by the acceptance policy of this section.

tokenizer treats as safe to commit: a whitespace-delimited lexical word in spacing scripts such as EN→DE/IT, or a single CJK character in EN→ZH, so partial subword fragments are never emitted.

4.5 Runtime Query/Key Capture

Equation (2) is only useful if the deployed runtime exposes the exact queries and keys consumed by attention. In the deployed vLLM path, those tensors never appear as ordinary Python-visible objects. We therefore install a per-layer observer, copy prompt keys and draft-row queries into fixed-shape buffers, and route capture through a custom-op-style call that survives runtime graph lowering. A zero-valued sentinel dependency keeps the observer in the transitive fan-in of the graph output:

$$\mathbf{o}'_\ell = \mathbf{o}_\ell + \Phi\big(\ell, \text{pos}, \mathbf{Q}^{(\ell)}, \mathbf{K}^{(\ell)}\big), \qquad \Phi \equiv \mathbf{0}. \quad (8)$$

Figure 5 separates the capture path into three stages. During setup, we patch the attention forward and install fixed-shape observer slots before the graph is captured. During each forward, the live observer records prompt keys and draft-row queries for the selected heads after the module's q/k normalization and rotary embedding, then returns $\Phi \equiv \mathbf{0}$ back into the attention output. After the forward, a short Python-side replay reconstructs only the selective draft-to-source block consumed by the policy and by the parity suite. The sentinel add-back of Eq. (8) is therefore only a liveness device: without the additive zero, graph lowering dead-code-eliminates the observer path and no usable Q/K buffers survive on the deployed runtime. Appendix D keeps the replay equations, parity measurements, and qualitative probe.

5 Results

5.1 Experimental Setup

The experiments run on a single NVIDIA A40. The MT component uses Gemma-4 E4B-it in bfloat16 through vLLM; the speech component uses Qwen3-ASR with forced alignment. In evaluation, we use the official IWSLT 2026 MCIF dev set (∼ 2.1 hours total, 21 long academic talks) resegmented by OmniSTEval. We report BLEU and chrF against the dev references, XCOMET-XL for translation adequacy (Guerreiro et al., 2024), and LongYAAL latency in its CU and CA variants (Polák et al., 2025). The streaming loop follows the Simulstream setting of unsegmented long-form audio with revision-capable evaluation (Gaido et al., 2025), but our cascade itself is synchronous: ASR and MT are serialized on one GPU so that policy effects are not confounded with scheduler overlap.

Section 5.5 compares three MT implementations on 16 text-only prompts: Transformers eager, Transformers SDPA *qk-fast*, and deployed vLLM *qk-fast*. All use greedy decoding with the same retained head set.

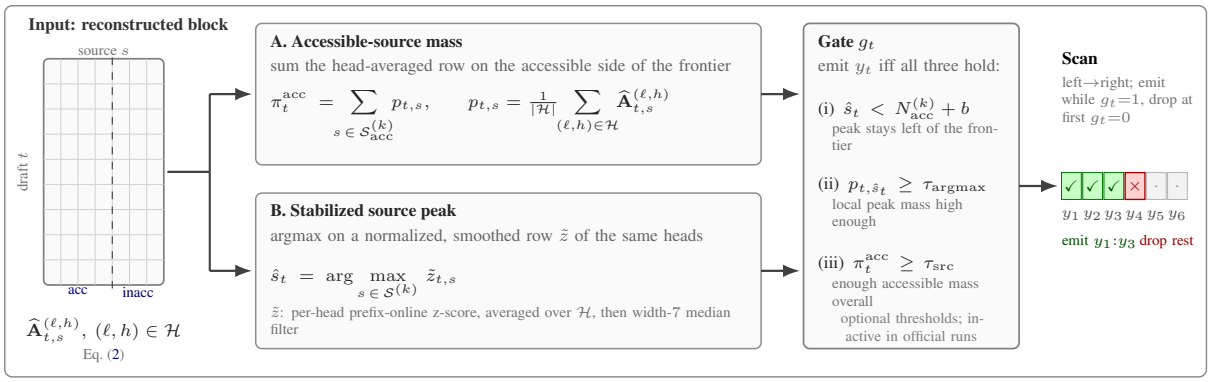

Figure 4: **From the reconstructed block to an acceptance decision.** The selective *qk-fast* reconstruction produces $\widehat{\mathbf{A}}^{(\ell,h)}_{t,s}$ for $(\ell, h) \in \mathcal{H}$, with draft positions t against source positions s (left). The policy reads it through two parallel aggregations. *Branch A* averages the selected heads into a row $p_{t,\cdot}$ and sums its mass on the accessible side of the frontier, giving the provenance score π_t^{acc}. *Branch B* returns the peak $\hat{s}_t$ of a stabilized row $\tilde{z}_{t,\cdot}$, where $\tilde{z}$ is obtained by z-scoring each head with prefix-online Welford moments, averaging over $\mathcal{H}$, and applying a width-7 median filter along the source axis. The gate g_t accepts draft token t iff the peak stays on the accessible side of the frontier, the peak mass exceeds τ_{argmax}, and the accessible-source mass exceeds τ_{src}. A left-to-right scan emits the longest run of accepting tokens and drops the remainder at the first failure.

5.2 System Results

Table 1 reports the two official operating points together with the organizers' no-context baselines for comparison. The pattern is consistent across EN→DE, EN→ZH, and EN→IT: the low-latency regime stays near a 2 s CU-LongYAAL, while the high-latency regime moves to a clearly stronger quality point at the expected latency cost. CA-LongYAAL is consistently below CU-LongYAAL here because OmniSTEval's CA mode replaces each chunk-boundary audio increment by the actual wall-clock increment spent processing that chunk. Since the deployed system runs faster than real time, the CA timestamps can be earlier than the CU chunk-boundary timestamps. We read Table 1 as evidence that the AlignAtt4LLM realization supports both a stable low-latency operating point and a higher-quality regime on the same decoder-only Gemma backend.

Against the supplied organizers' no-context baselines, AlignAtt4LLM clearly wins the < 2 s and < 4 s latency regimes for EN→DE and EN→IT. EN→ZH is mixed: our system reaches comparable chrF at high latency and slightly higher chrF at low latency, but the organizers' baseline remains ahead on BLEU and XCOMET-XL. The offline diagnostic rows show substantial backbone headroom once online commitment is removed. They are not latency-comparable to the streaming runs, but they help separate model capacity from the cost of the simultaneous policy. Appendix A.1 gives the minimal setup details. Gemma-4 is not necessarily the optimal MT backbone for this policy: recent translation-focused decoder-only LLMs such as HY-MT-1.5 (Zheng et al., 2025) and MiLMMT-46 (Shang et al., 2026) achieve strong multilingual results, especially for Chinese, and require no changes to the AlignAtt implementation, though

Lang.	System	Reg.	BLEU	chrF	XCOM.	L. YAAL CU	L. YAAL CA
en→de	Baseline	low	22.35	56.7	0.748	**1.81**	n/a
	Ours	low	**28.76**	**62.1**	**0.875**	2.00	1.63
	Baseline	high	26.31	59.2	0.819	**2.63**	n/a
	Ours	high	**32.63**	**64.2**	**0.902**	3.53	3.14
	Offline	-	*38.57*	*67.1*	*0.938*	*n/a*	*n/a*
en→zh	Baseline	low	**40.85**	34.1	**0.750**	**1.91**	n/a
	Ours	low	36.01	**35.0**	0.743	1.95	1.77
	Baseline	high	**43.85**	**37.8**	**0.795**	3.48	n/a
	Ours	high	39.86	**37.8**	0.778	**3.27**	3.09
	Offline	-	*48.53*	*43.4*	*0.848*	*n/a*	*n/a*
en→it	Baseline	low	30.63	62.0	0.683	**1.76**	n/a
	Ours	low	**40.10**	**68.0**	**0.805**	1.98	1.62
	Baseline	high	37.28	65.4	0.781	**3.30**	n/a
	Ours	high	**44.46**	**70.1**	**0.841**	3.48	3.10
	Offline	-	*49.88*	*73.0*	*0.895*	*n/a*	*n/a*

Table 1: **Cascade results on the IWSLT 2026 dev set** (21 clips, OmniSTEval long-form resegmentation). Ours low rows use the low-latency setting with $\Delta_{\mathrm{chunk}} = 850$ ms; Ours high rows use $\Delta_{\mathrm{chunk}} = 1500$ ms. Organizer baseline rows are the no-context baseline outputs provided with CU-LongYAAL; their CA-LongYAAL was not available. Offline rows are diagnostic quality-only cascades with Qwen full-audio ASR followed by Gemma final-mode MT; they have no streaming latency score.

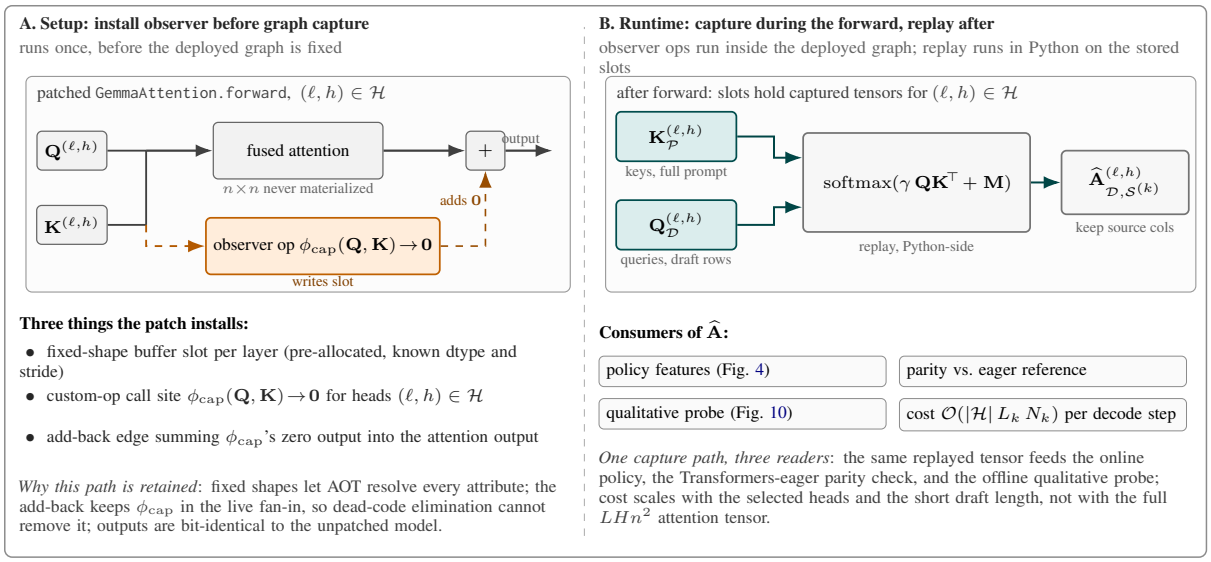

Figure 5: **Observer lifecycle on the deployed vLLM path.** *Left (A), once, before graph capture:* the forward of `GemmaAttention` is patched so that for each selected head $(\ell, h) \in \mathcal{H}$ the query and key tensors also pass through an observer custom op ϕ_{cap} (dashed orange side path). The op writes the selected slices into fixed-shape, pre-allocated slots and returns a zero tensor that is added back into the attention output; this keeps the op in the live fan-in of the graph so graph optimization cannot dead-code-eliminate it, while the numerical output of the layer is unchanged. *Right (B), every forward:* after the forward finishes, the observer slots hold $\mathbf{K}_{\mathcal{P}}^{(\ell,h)}$ for the full prompt and $\mathbf{Q}_{\mathcal{D}}^{(\ell,h)}$ for the draft rows; a short Python-side replay reapplies the module scaling and the same prompt/suffix masks used by the deployed attention kernel, then keeps the source columns to yield the selective block $\widehat{\mathbf{A}}_{\mathcal{D},\mathcal{S}^{(k)}}^{(\ell,h)}$ that feeds the policy, the parity check against a Transformers reference, and the qualitative probe.

the alignment head set and acceptance thresholds would need to be recalibrated for each new backbone.

5.3 Additional Analysis: ASR Tail Reliability

The official runs expose each Qwen3 ASR word to MT as soon as its aligned end time is observed. A post-hoc reliability analysis suggests a slightly more conservative default for future runs: clip the last 250 ms of the live ASR tail before passing the source prefix to MT. This is a policy-level timing margin rather than a lexical repair. Because the Qwen path re-transcribes the live utterance tail from scratch at each chunk, the final ASR words are exactly the part most likely to be revised on the next chunk. This instability is partly masked by AlignAtt, which often stops MT before translating the newest source words near the frontier. Reducing tail noise could therefore allow a less conservative acceptance policy, not only a longer hold-back. On the 21 MCIF dev talks, the reference error rate of words at the current Qwen3 ASR tail end is 17.1%, but drops to 8.3% at 250 ms and then remains nearly flat. We therefore keep the reported numbers unchanged, but recommend a 250 ms ASR-tail hold-back as the maintained default going forward. Appendix A gives the comparison with the alternative ASR component parameters.

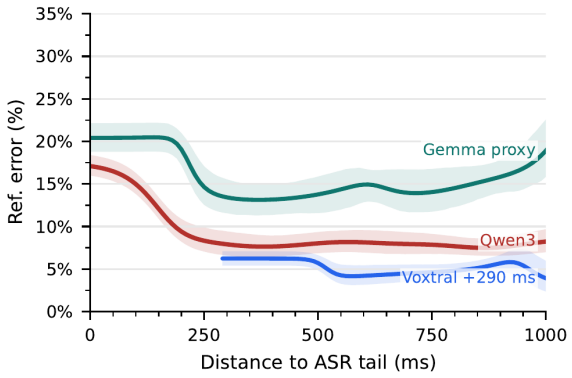

Figure 6: **Live-tail ASR reference error.** Reference-error rate by distance to the current ASR tail; bands are 90% audio-bootstrap intervals. Qwen3 drops from 17.1% at the tail to 8.3% at 250 ms and then stays flat. Voxtral is shifted by +290 ms CU-LongYAAL; Gemma remains unshifted because its timestamps/LongYAAL are unreliable under prompt leakage.

Setting	XCOMET-XL ↑	CU (s) ↓	CA (s) ↓
Top-8 heads	0.879	**1.96**	**1.65**
All 336 heads	**0.885**	2.06	1.83

Table 2: **Auxiliary EN→DE MT head-set comparison at $\Delta_{\text{chunk}} = 1100$ ms.** Both rows use the same maintained runtime; only the MT head set changes. End-to-end quality is nearly equivalent, but all-head replay increases CU and especially CA because the observer becomes less selective while the runtime must reconstruct much more attention at each MT step.

5.4 Head Filtering Matters End-to-End

Table 2 is an auxiliary EN→DE rerun included only to isolate the MT head-set effect under the maintained *qk-fast* runtime. End-to-end quality is nearly equivalent, but the latency split is informative. CU-LongYAAL increases by 100.3 ms, which likely means the all-head observer is slightly less precise and therefore a bit more generous on tokens whose support extends beyond the currently available source prefix. CA-LongYAAL rises more strongly, by +179.5 ms, because replaying all 336 heads forces the runtime to reconstruct much more attention state at each MT step. We therefore read the auxiliary comparison as evidence that a small retained head set captures most of the useful MT alignment signal at much lower runtime cost.

5.5 Runtime Cost of the MT Component

The MT component must be computationally cheap enough to justify deployment. Figure 7 shows that the vLLM replay path is much cheaper than the minimal eager implementation we use as an inspection reference: median cost drops from 63.7 to 25.4 ms/token, while the Transformers SDPA *qk-fast* replay remains close to the eager baseline at 59.0 ms/token. This confirms that the MT contribution is not merely inspectable in principle; it is practical inside the runtime we actually deploy.

6 Conclusion

We presented a Qwen3 ASR + Gemma-4 MT cascade for simultaneous speech translation, and its key technical ingredient is an AlignAtt policy adapted to decoder-only MT. Deterministic prompt layout, offline head selection, selective attention replay, and runtime query/key capture make the MT policy usable on the deployed vLLM path. On the dev set, this yields a low-latency operating point near 2 s CU-LongYAAL and a high-latency regime that is roughly 1.5 s slower but clearly stronger in

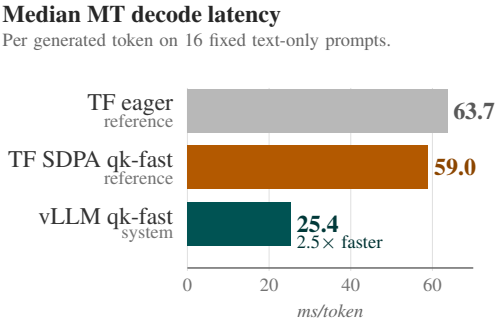

Figure 7: **Inference-time comparison of MT capture implementations.** Median latency per generated token on a fixed 16-prompt text-only suite, for a minimal Transformers eager reference, a Transformers SDPA *qk-fast* reference that reconstructs source rows from captured layer inputs, and the deployed vLLM *qk-fast* path used by the presented system.

BLEU and XCOMET-XL. Compared with the supplied organizers' no-context baselines, the system clearly wins the < 2 s and < 4 s latency regimes for English to German and Italian, while English to Chinese remains mixed: chrF is competitive, but BLEU and XCOMET-XL still favor the baseline. This appears partly tied to the Gemma-4 MT backbone, whose Chinese generations were weaker in our runs; because AlignAtt4LLM only requires a deterministic prompt layout, calibrated heads, and Q/K capture, the same policy can be ported to other decoder-only LLMs, making translation-focused backbones such as HY-MT-1.5 (Zheng et al., 2025), MiLMMT-46 (Shang et al., 2026), or a Qwen-family MT backbone, natural next targets for EN→ZH.

Acknowledgements

This work was supported by Czech Operational Program OP JAK, the MSCA CZ project MSCA Fellowships – Charles University 4, CZ.02.01.01/00/22_010/0013392, "LCT".

References

David Ifeoluwa Adelani, Victor Agostinelli, Antonios Anastasopoulos, 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, Antoine Laurent, and 33 others. 2026. Speech translation and metrics in 2026: Findings of the IWSLT campaign. In *Proceedings of the 23rd International Conference on Spoken Language Trans-*

lation (IWSLT 2026), San Diego, California, US. Association for Computational Linguistics.

Marco Gaido, Sara Papi, Mauro Cettolo, Matteo Negri, and Luisa Bentivogli. 2025. Simulstream: Open-source toolkit for evaluation and demonstration of streaming speech-to-text translation systems. *Preprint*, arXiv:2512.17648.

Google DeepMind. 2026. Gemma 4 E4B-it model card. Hugging Face model card.

Nuno M. Guerreiro, Ricardo Rei, Daan van Stigt, Luisa Coheur, Pierre Colombo, and André F. T. Martins. 2024. xCOMET: Transparent machine translation evaluation through fine-grained error detection. *Transactions of the Association for Computational Linguistics*, 12:979–995.

Tom Kocmi, Ekaterina Artemova, Eleftherios Avramidis, Rachel Bawden, Ondřej Bojar, Konstantin Dranch, Anton Dvorkovich, Sergey Dukanov, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, Barry Haddow, Marzena Karpinska, Philipp Koehn, Howard Lakougna, Jessica Lundin, Christof Monz, Kenton Murray, and 10 others. 2025. Findings of the WMT25 general machine translation shared task: Time to stop evaluating on easy test sets. In *Proceedings of the Tenth Conference on Machine Translation*, pages 355–413. Association for Computational Linguistics.

Tom Kocmi, Eleftherios Avramidis, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, Barry Haddow, Marzena Karpinska, Philipp Koehn, Benjamin Marie, Christof Monz, Kenton Murray, Masaaki Nagata, Martin Popel, Maja Popović, and 3 others. 2024. Findings of the WMT24 general machine translation shared task: The LLM era is here but MT is not solved yet. In *Proceedings of the Ninth Conference on Machine Translation*, pages 1–46. Association for Computational Linguistics.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with PagedAttention. *Preprint*, arXiv:2309.06180.

Binbin Liu, Wenhan Han, Feng Chen, Yifan Zhang, Ping Guo, Haobin Lin, Bingni Zhang, Taifeng Wang, and Yin Zheng. 2026. Token alignment heads: Unveiling attention's role in LLM multilingual translation. ICLR 2026 poster, OpenReview.

Dominik Macháček and Peter Polák. 2025. Simultaneous translation with offline speech and LLM models in CUNI submission to IWSLT 2025. In *Proceedings of the 22nd International Conference on Spoken Language Translation (IWSLT 2025)*, pages 389–398, Vienna, Austria. Association for Computational Linguistics.

OpenAI. 2025. Introducing GPT-5 for developers. OpenAI blog.

Sara Papi, Marco Gaido, Matteo Negri, and Luisa Bentivogli. 2024. StreamAtt: Direct streaming speech-to-text translation with attention-based audio history selection. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3692–3707, Bangkok, Thailand. Association for Computational Linguistics.

Sara Papi, Marco Turchi, and Matteo Negri. 2023. AlignAtt: Using attention-based audio-translation alignments as a guide for simultaneous speech translation. In *Proceedings of Interspeech 2023*, pages 3974–3978.

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.

Peter Polák, Sara Papi, Luisa Bentivogli, and Ondřej Bojar. 2025. Better late than never: Meta-evaluation of latency metrics for simultaneous speech-to-text translation. *Preprint*, arXiv:2509.17349.

Peter Polák, Ngoc-Quan Pham, Tuan Nam Nguyen, Danni Liu, Carlos Mullov, Jan Niehues, Ondřej Bojar, and Alexander Waibel. 2022. CUNI-KIT system for simultaneous speech translation task at IWSLT 2022. In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, pages 277–285, Dublin, Ireland (in-person and online). Association for Computational Linguistics.

Peter Polák, Brian Yan, Shinji Watanabe, Alex Waibel, and Ondřej Bojar. 2023. Incremental Blockwise Beam Search for Simultaneous Speech Translation with Controllable Quality-Latency Tradeoff. In *Proc. INTERSPEECH 2023*, pages 3979–3983.

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.

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

Xian Shi, Xiong Wang, Zhifang Guo, Yongqi Wang, Pei Zhang, Xinyu Zhang, Zishan Guo, Hongkun Hao, Yu Xi, Baosong Yang, Jin Xu, Jingren Zhou, and Junyang Lin. 2026. Qwen3-ASR technical report. *Preprint*, arXiv:2601.21337.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. 2021. RoFormer: Enhanced transformer with rotary position embedding. *Preprint*, arXiv:2104.09864.

Mao Zheng, Zheng Li, Tao Chen, Mingyang Song, and Di Wang. 2025. HY-MT1.5 technical report. *Preprint*, arXiv:2512.24092.

A Additional ASR Analysis

The analysis below justifies the source ASR front end used by the cascade and the recommended source-tail default for future runs.

ASR front-end selection. We tested three ASR front ends during development: Qwen3-ASR with the Qwen3 forced aligner, Voxtral Realtime 4B, and a direct Gemma E4B ASR local-agreement probe. Table 3 summarizes the results on the the MCIF dev set. Voxtral gives the lowest final WER, but it is not real-time in our serialized single-GPU loop and its CU-LongYAAL is higher than Qwen3. It also advances the transcript with a more regular lag, whereas Qwen3 often exposes compact multi-word updates, which better matches our chunk-synchronous MT policy. Gemma E4B would be attractive as a single-model substrate, but its WER, prompt leakage on long talks, and unstable raw latency make it unsuitable here. We therefore retain Qwen3 forced ASR.

ASR front end	WER ↓	CU ↓	RTF ↓	Decision
Qwen3 forced	8.91	0.87 s	0.34	keep
Voxtral RT 4B	7.15	1.16 s	1.34	reject
Gemma E4B LA	16.69	∼1.8 s[†]	0.51	reject

Table 3: **ASR front-end comparison on 21 MCIF dev talks.** WER is percent and CU is CU-LongYAAL. [†]Gemma's raw 0.32 s LongYAAL is prompt-leak-contaminated; clipping or discarding negative emissions gives about 1.8 s.

A.1 Offline Cascade Diagnostic

As a development tool that brackets the streaming numbers from above, we also ran an offline cascade on the same 21 MCIF dev talks: Qwen3-ASR was applied once to each complete audio file, then Gemma-4 was run in final mode on sentence-level chunks resegmented as in the streaming evaluation. The full-audio ASR transcripts have 7.40% weighted corpus WER against the English reference (90% audio-bootstrap interval $[6.27, 8.59]$), so the offline run sees source much closer to the reference than any online chunk. Differences between offline and streaming output therefore isolate the cost of the AlignAtt commit policy and the live ASR tail from the intrinsic quality of the backbone: drops on EN→ZH that already appear here point to Gemma-4 rather than to the simultaneous policy.

B Synchronization Modes

Figure 8 summarizes alternative synchronization modes.

Our cascade loads the ASR and MT models as two separate blocking `vllm.LLM` instances on a single GPU, which serializes every audio chunk into an ASR decode followed by an MT decode (Figure 8c). This conservative regime keeps policy effects separate from scheduler overlap. Alternative pipelined or fully asynchronous regimes would mainly reduce CA-LongYAAL at fixed CU-LongYAAL, but would require a non-blocking scheduler and a more careful treatment of stale MT requests. We leave that scheduler axis to future work.

C MT Alignment-Head Diagnostics

See Figure 9 and Table 4.

Pair	Top-8 TS	All-336 TS	Gain	Aligned tokens
EN→DE	90.40	68.49	**+21.91**	11,209
EN→ZH	93.48	65.79	**+27.70**	7,582
EN→IT	91.90	67.42	**+24.49**	12,056

Table 4: **MT head-set filtering ablation on held-out word-aligned dev examples.** Scores are reported in points (100 × TS) against gold aligned source tokens.

D Observer Replay and Qualitative Diagnostics

Below we report the prompt/suffix replay equations, one qualitative probe, and the numerical parity measurements on the deployed vLLM path.

D.1 Prompt-space replay details

After capture, we replay the draft rows against two key blocks: the pre-draft prompt positions $\mathcal{P}^{(k)}$ and the current draft positions $\mathcal{D}^{(k)}$, which form the autoregressive suffix. Writing $\mathbf{B}_{\star}^{(\ell,h)} := \mathbf{B}_{\star}^{(\ell)}[h]$ for head-specific captured tensors, $\gamma^{(\ell)}$ for the attention module's runtime scaling factor, and $\mathbf{M}_{\text{prompt}}^{(\ell,h)}$ / $\mathbf{M}_{\text{draft}}^{(\ell,h)}$ for the corresponding sliding-window and causal/window masks, we first form the two logit blocks

$$\mathbf{P}^{(\ell,h)} = \gamma^{(\ell)} \mathbf{B}_{\text{dQ}}^{(\ell,h)} \left(\mathbf{B}_{\text{pK}}^{(\ell,h)}\right)^{\top} + \mathbf{M}_{\text{prompt}}^{(\ell,h)}, \tag{9}$$

$$\mathbf{R}^{(\ell,h)} = \gamma^{(\ell)} \mathbf{B}_{\text{dQ}}^{(\ell,h)} \left(\mathbf{B}_{\text{dK}}^{(\ell,h)}\right)^{\top} + \mathbf{M}_{\text{draft}}^{(\ell,h)}, \tag{10}$$

$$\widetilde{\mathbf{A}}_{\mathcal{D},\,\mathcal{P}\cup\mathcal{D}}^{(\ell,h)} = \text{softmax}_{\text{row}}\left(\left[\mathbf{P}^{(\ell,h)} \| \mathbf{R}^{(\ell,h)}\right]\right). \tag{11}$$

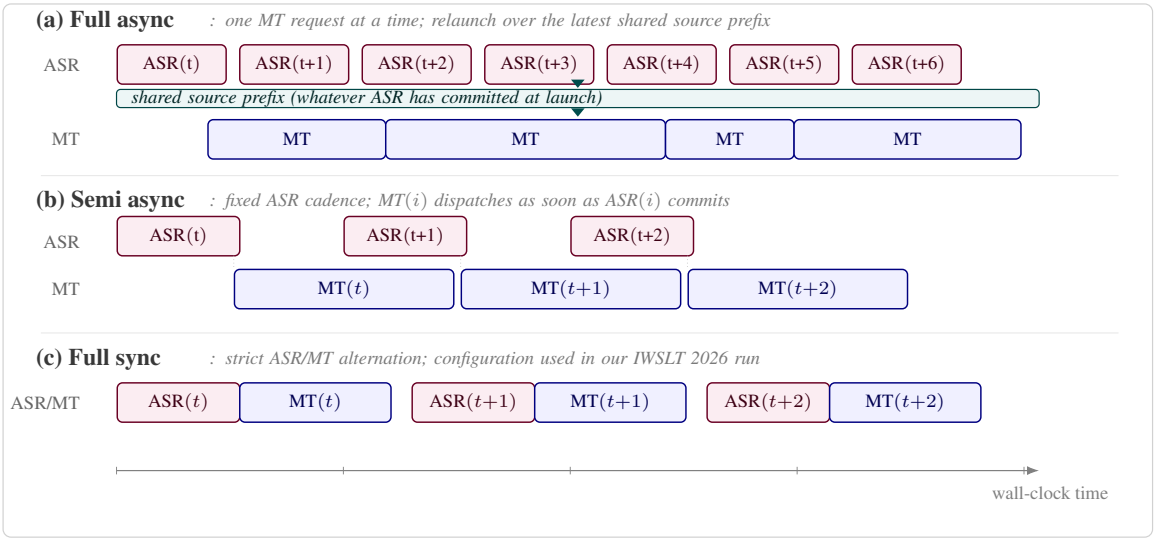

Figure 8: **Synchronization regimes for ASR and MT sharing one GPU.** Regime (c) is the deployed IWSLT schedule; regimes (a) and (b) illustrate less-blocking alternatives that require asynchronous request handling.

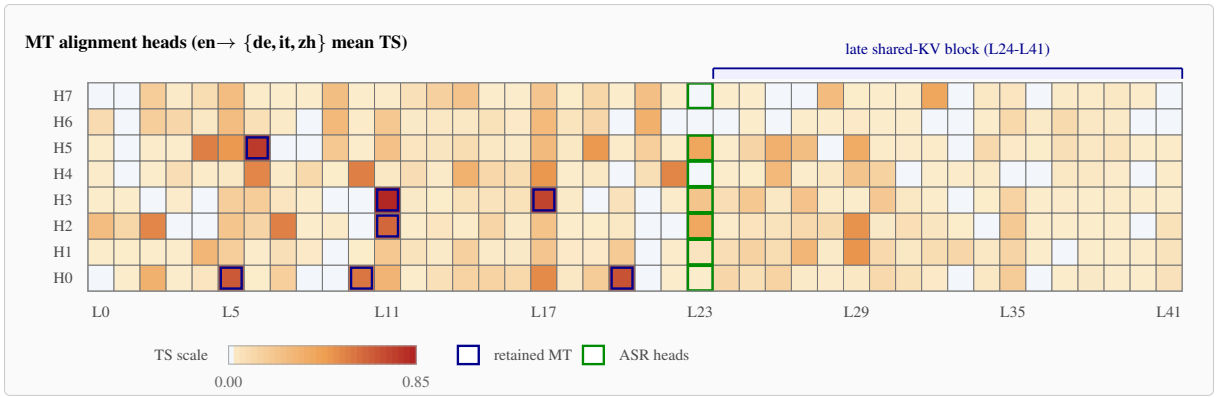

Figure 9: **Architecture-aware view of retained MT alignment heads.** Retained MT heads are sparse, late, and only partly overlap with the ASR set.

The row-wise softmax is taken over the concatenated prompt and draft columns, so source positions compete with all other causal context exactly as in the deployed attention row. Restricting $\widetilde{\mathbf{A}}^{(\ell,h)}_{\mathcal{D},\mathcal{P}\cup\mathcal{D}}$ to the source columns $\phi^{(k)}(s)$ recovers the policy-visible block of Eq. (2). The replay cost therefore scales with the selected head set and the short draft length, not with the full LHn^2 attention tensor.

D.2 Qualitative reconstruction example

Figure 10 shows the full reconstruction stack on an EN→DE text-only MT probe: the exact prompt partition seen by the decoder, word-level aggregation of the replayed rows for readability, and the four-way provenance accounting for every drafted word.

The green band in Figure 10 is previously committed target text reused as causal context; it is not itself the object of the gate. The orange words are the current draft scanned left-to-right by the policy. We allow $\hat{s}_t$ to move backward within a live draft when the decoder revisits earlier source material under reordering, which is why the residual prompt and suffix masses remain explicit in the decoder-only formulation.

D.3 Numerical parity

Let $\mathbf{A}^{(\mathrm{TF})}$ denote the reference attention tensor produced by a Transformers reference forward on the identical prompt, and let $\Delta\mathbf{A} = \widetilde{\mathbf{A}} - \mathbf{A}^{(\mathrm{TF})}$. On a curated parity set, the reference and deployed paths make bit-identical acceptance decisions and satisfy

$$\begin{aligned} \|\Delta\mathbf{A}\|_\infty &\le 1.2 \times 10^{-2}, \\ \|\Delta\mathbf{A}\|_1/(nL_k) &\le 4 \times 10^{-4}. \end{aligned} \tag{12}$$

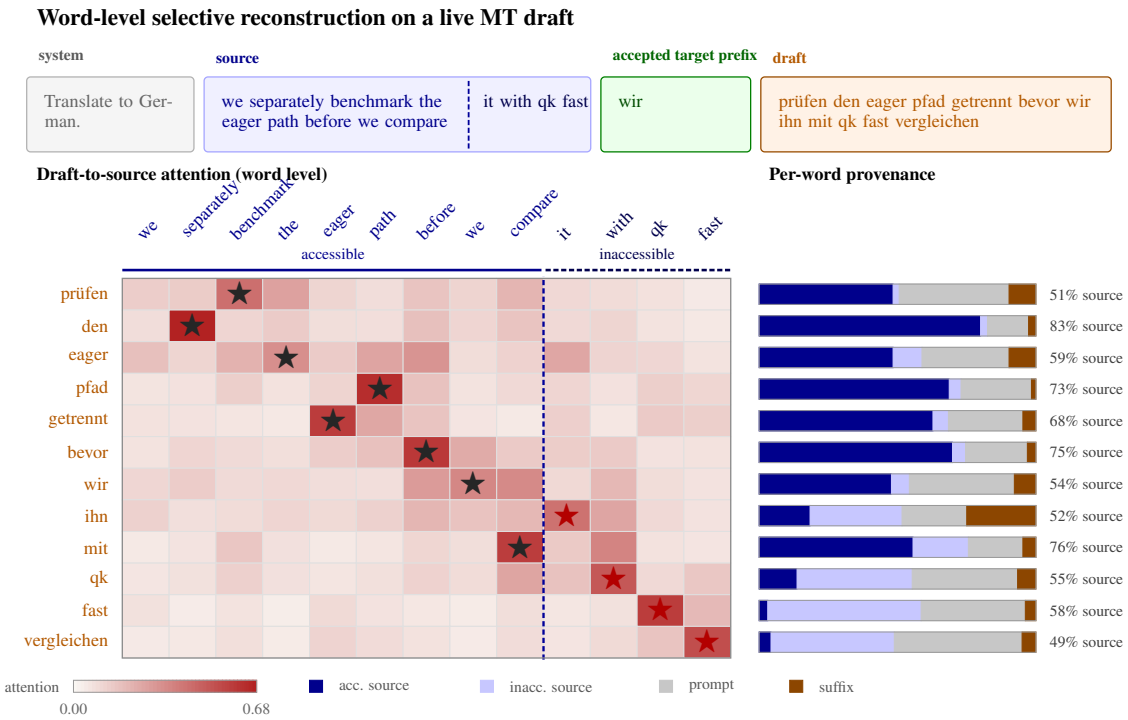

Figure 10: **Word-level selective reconstruction on a live MT draft.** *Top ribbon:* prompt partition into system instruction, live source, accepted target prefix, and current draft. *Left panel:* reconstructed draft-to-source attention from the selected AlignAtt heads, aggregated to words and split by the dashed accessibility frontier; black stars stay on the accessible side, while the first red star marks the SOURCE-FRONTIER failure. *Right panel:* each drafted word is decomposed into accessible-source, inaccessible-source, prompt, and suffix mass, with the numeric label giving the total source share.