Babelbit: A Bittensor Subnet for Human-Centred Low-Latency Speech Translation

Prediction, paraphrase, speech-mode transformers, and incentive-driven optimisation

Matthew Karas
Founder, Babelbit Ltd.
https://babelbit.ai
mailto:info@babelbit.ai

v1.0: May 8, 2026

Abstract

Simultaneous achine translation has traditionally been optimised for translation accuracy and speed. That framing is insufficient for real-time spoken conversation, where the decisive quantity is how quickly a listener can receive enough meaning to continue the interaction naturally. Professional interpreters solve this human-centred problem by paraphrasing, compressing, omitting disfluencies, and anticipating predictable completions. Babelbit applies the same principle to machine translation, with advantages unavailable to human interpreters: transformer-scale predictive modelling, direct speech tokenisation, parallel evaluation at population scale, and programmable confidence thresholds. This whitepaper defines the Babelbit Bittensor subnet as a research and production mechanism progressing from an initial text-only utterance-completion challenge to a real-time speech-to-speech translation system. We formalise *early adequacy*, introduce metrics for phrase-level latency, describe a speech-mode transformer architecture capable of prediction, translation, paraphrasing, and speech generation in a one-shot process, and argue that Bittensor is a natural environment for iterative, feature-specific optimisation. The central claim is that the next frontier in translation is not faster literal translation, but end-user adequacy delivered early, clearly, politely, and safely.

Summary

- **Core proposition:** Real-time translation should optimise for *conversational adequacy*—the earliest moment at which a listener can act—rather than literal fidelity at sentence end.
- **Bittensor:** The subnet provides a decentralised, incentive-driven environment for iteratively improving prediction, paraphrase, safety, domain adaptation, and speech-mode modelling across many language pairs.
- **Roadmap:** Challenges progress from text-only utterance completion through basic speech-to-speech translation to ongoing feature extension.

1 The problem: latency is not legacy speed-up

Babelbit is a Bittensor subnet that optimises multilingual speech translation for *conversational adequacy under latency pressure*: the goal is to deliver the right meaning to the listener at the earliest safe moment, not merely to reproduce the source utterance as fast as possible.

The standard engineering view of speech translation decomposes the problem into automatic speech recognition, text translation, and speech synthesis. Modern systems can execute these steps far faster than real time once the relevant input is available. The bottleneck in live conversation is different: the system often does not yet know what it is safe to say. In many language pairs, key information may arrive late in the clause or sentence. The familiar interpreter's problem of "waiting for the verb" is not an anecdotal curiosity, but a structural reason why literal simultaneous translation is hard.

Babelbit begins from a different objective. The target is not to reproduce the whole input faster; it is to determine when the source utterance has become *adequate enough* for a useful target-language act. This is especially important because conversation is not a sequence of finalised documents. It is a stream of social actions: agreeing, objecting, clarifying, greeting, hedging, asking, warning, apologising. In many cases, the correct conversational act is obvious before the source speaker has finished their sentence.

A literal system may wait until it can produce a complete, sentence-level translation. A Babelbit-optimised system may instead commit earlier to a shorter target-language act when the speaker's intent is already clear. This is not a lower-quality translation in the conversational sense. It is a better interaction if it preserves the speaker's intent, reduces latency, and avoids unnecessary verbal bulk.

The design therefore separates three concepts that are often conflated:

1. **Computational speed**: how quickly a model can process available input.
2. **Recognition delay**: how long the system must wait before the intended meaning is predictable.
3. **Interactional latency**: how long the listener waits before receiving a usable conversational signal.

Babelbit attacks the second and third quantities directly. Speed is still important, but it is only one component of the user experience.

2 Why now: speech-mode language models

The timing is important. Until recently, the obvious path to speech translation was a cascade: speech to text, text translation, then text to speech. That approach inherits latency at every boundary. It also restricts the system to a text bottleneck, losing prosody, hesitation, speaker identity, and other cues that matter for prediction and interpretation.

Recent research has begun to collapse the orthodox distinction between natural language processing and speech technology. Large multimodal and speech-aware language models can consume or emit speech representations, including discrete audio tokens or continuous speech features. Meta's SeamlessM4T work demonstrates a unified multilingual speech and text translation system across many modalities and languages. SeamlessStreaming and SeamlessExpressive extend that agenda toward streaming and expressivity. Kyutai's Hibiki introduces a decoder-only multistream model that synchronously processes source and target speech and jointly produces text and audio tokens for simultaneous translation. SimulS2S-LLM shows how speech LLMs can be adapted for simultaneous speech-to-speech inference using boundary-aware prompts, discrete output speech tokens, and an incremental beam search policy. InfiniSST and the CMU IWSLT 2025 system similarly reflect the shift toward LLM-based streaming speech translation and computation-aware latency.

Along with Babelbit, these systems are establishing a new field. Babelbit's claim, however, is that the most valuable next step is not merely to produce a better streaming version of literal translation. The subnet should optimise for aggressive prediction, human-centred paraphrasing, safety, politeness, domain-specific terminology, and customer-specific lexicons. In other words, the substrate has arrived: speech can increasingly be represented in token-like forms and be manipulated by speech mode language-model architectures. The product frontier is now the objective function.

3 Human interpreting as the design model

Professional interpreters do not behave like literal, lossless, low-latency text translators. They routinely compress (or expand), reorder, paraphrase, and omit material that is redundant, rude, distracting, or pragmatically irrelevant. They also make early commitments when the continuation is clear. A human interpreter may hear the beginning of a formulaic phrase, identify the intended

act, and begin rendering the target language before the source phrase is complete.

In fact it is not uncommon for an interpreter to achieve negative latency, when benchmarked correctly, i.e. by phrase completion. One clear example of this is in stock responses where even the start of a phrase is known before it is uttered:

Greeting: *as-salam alaykum*

Response: *wa alaykum salam*

This matters because the success criterion is not lexical equivalence. It is whether the target listener receives the meaning, tone, and social force needed to participate in the conversation. For example, the system should be rewarded for removing interjections and repetitions, turning vague speech into clear propositions, preserving politeness where required, and filtering or softening expletives, blasphemy, racism, or other prejudicial material when the context and product policy require it. These behaviours are not defects; they are part of what interpreters are trained to do.

Computers also have an advantage over human interpreters, so as the technology matures, it is expected that in some respects it will be superior to human interpreters. Humans must train for years to speak while continuing to listen, and working memory is a limiting factor. A transformer-based speech system can maintain multiple hypotheses, track acoustic and semantic state, run a low-latency stream and a high-accuracy stream in parallel, and revise the record without forcing the live listener to wait. The engineering challenge is not whether a machine can keep listening while speaking. It can. The challenge is deciding *when* to speak and *what form* the output should take.

4 From text-only launch challenge to real-time speech-to-speech

The Babelbit subnet is moving from a text-only prediction challenge to a real-time speech-to-speech translation challenge. The text-only phase was not a detour. It isolated central questions: can a model predict the adequate completion of an utterance earlier than a literal translation pipeline would ordinarily commit? Can a network be trained to perform this prediction using open competitions on the Bittensor platform.

The initial mining challenge therefore asks miners to produce high-quality completions from prefixes. It does not require the miner to know whether its own output is adequate; validators can score the output at every prefix against the full utterance. This distinction is important. Training a predictor and training a confidence policy are related but separable tasks. The first creates value even before the second is solved.

The speech-to-speech phase adds three further layers:

1. Speech input must be tokenised with acoustic, lexical, and prosodic cues.
2. Output is speech, so latency must include generation.
3. The system avoids a text bottleneck by modelling source and target speech representations directly.

The target is a one-shot speech-mode transformer process that performs prediction, paraphrasing, translation, and speech generation within a unified architecture. However, the competition is not prescriptive. If miners come up with alternative architectures, which out-perform the supplied base-script, the subnet can pivot towards the optimal architecture, releasing it for other miners to build upon.

5 Review of Prediction Gains from Phase 1

5.1 Formalising early adequacy

While the trained model is not programmatically implementing the formal descriptions of adequacy, confidence, earliness etc, it was important to represent these with algebraic precision in order to

create the scoring mechanisms, which control the competition.

Let the source utterance be a sequence

$$X = (x_1, x_2, \ldots, x_N). \tag{1}$$

After observing a prefix

$$X_{1:k} = (x_1, \ldots, x_k), \tag{2}$$

the system proposes a candidate set

$$C_k = \{c_{k,1}, \ldots, c_{k,m_k}\}, \tag{3}$$

with a confidence distribution $q_k(c)$. The top candidate is

$$\hat{c}_k = \arg\max_{c \in C_k} q_k(c), \quad p_k = q_k(\hat{c}_k). \tag{4}$$

Next-word prediction models the local distribution

$$P(x_{k+1} \mid X_{1:k}). \tag{5}$$

Babelbit is concerned with global continuation and conversational adequacy:

$$P(X_{k+1:N} \mid X_{1:k}), \tag{6}$$

not because the system must reproduce the exact remaining words, but because it must decide whether it has inferred enough meaning to act.

Define lexical similarity $L(c, X) \in [0, 1]$ and semantic or judged adequacy $S(c, X) \in [0, 1]$. A simple adequacy function is

$$A(c, X) = \alpha L(c, X) + (1 - \alpha) S(c, X), \quad \alpha \in [0, 1]. \tag{7}$$

In deployment, S may be replaced or supplemented by an LLM-as-judge J_θ:

$$A(c, X) = \alpha L(c, X) + (1 - \alpha) J_\theta(c, X). \tag{8}$$

The judge is asked not whether the candidate is a literal translation, but whether it is adequate to keep the conversation going, preserving meaning units, intent, register, politeness, and safety.

The **Earliest Adequate Translation Point** (EATP) is

$$k^* = \min\{k \in \{1, \ldots, N\} : \exists c \in C_k \text{ s.t. } A(c, X) \geq \tau\}, \tag{9}$$

where τ is the threshold for conversational adequacy.

The normalised recognition lead is

$$\text{Lead}_\tau = \frac{N - k^*}{N - 1}. \tag{10}$$

A lead of one indicates adequacy from the first token; a lead of zero indicates adequacy only at the end.

An **Adequate Commitment Score** rewards earliness, confidence, and adequacy margin:

$$\text{ACS}_\tau = \frac{N - k^*}{N - 1} \cdot p^* \cdot \frac{A(c^*, X) - \tau}{1 - \tau}. \tag{11}$$

For smoother training, an early-weighted expected adequacy score can be used:

$$\text{EA}_\gamma = (1 - \gamma) \sum_{k=1}^{N} \gamma^{k-1} \sum_{c \in C_k} q_k(c) A(c, X), \quad 0 < \gamma < 1. \tag{12}$$

These metrics shift evaluation from sentence-final correctness to phrase-level usable meaning.

5.2 Prediction confidence and early commitment

Babelbit distinguishes *being able to make a good early prediction* from *knowing that the prediction is good enough*. The first subnet phase can reward predictors directly. The production system must also learn a calibrated decision policy.

A raw LLM confidence score is not sufficient. Models can output numeric self-confidence, but these numbers are not reliably calibrated. A practical confidence model should combine token probabilities, entropy, beam or sample agreement, semantic stability, ASR stability, source-side prosody, prefix length, and domain priors. For a prefix k, log a tuple

$$(p_k, f_k, y_k), \quad y_k = \mathbf{1}\{A(\hat{c}_k, X) \geq \tau\}, \tag{13}$$

where f_k denotes additional features such as entropy, margin, beam agreement, semantic clustering tightness, ASR confidence, and recent hypothesis stability. A monotone calibrator estimates

$$\hat{P}_k \approx \Pr(y_k = 1 \mid p_k, f_k). \tag{14}$$

The target-quality policy commits at the earliest prefix satisfying

$$k^*_{\text{commit}} = \min\{k : \hat{P}_k \geq q\}, \tag{15}$$

where q is set by context. Small talk may accept a lower threshold; medical, legal, diplomatic, or industrial safety settings require a higher one.

Predictability is context-dependent. A phrase such as “may the force...” is highly predictable. A phrase such as “my favourite poems are...” is not.

This is not a hand-coded calculation, to be implemented. Rather it describes and calibrates the target of network training. The subnet should therefore reward both aggressive early prediction and selective abstention. In the live product, a good system speaks early when the continuation is stable and waits when it is not, because it is trained to do so by example, not because it performs these calculations as each output token is generated.

6 User experience is not literal translation accuracy

Literal accuracy is not the same as user utility. A system can be lexically faithful and still unusable if it arrives late, repeats filler, preserves offensive material unnecessarily, or buries the speaker’s point in verbal clutter. Conversely, a short paraphrase can be more faithful to the communicative act than a word-for-word rendering.

Babelbit therefore treats user experience as a bundle of measurable but partly independent qualities:

1. **Meaning delivery**: the target listener receives the intended act as soon as safely possible.
2. **Brevity**: the output removes repetition, false starts, and low-value filler.
3. **Clarity**: the output extracts the point from vague or rambling speech.
4. **Politeness and cultural safety**: the output adapts register and avoids unnecessary expletives, blasphemy, racism, or prejudicial phrasing.
5. **Domain fit**: terminology reflects the customer context, such as medical, legal, industrial, EU, UN, or corporate lexicons.
6. **Interactional smoothness**: the listener experiences natural turn-taking rather than delayed bursts.

These criteria require a benchmark that is not reducible to BLEU or word error rate. BLEU-like and COMET-like metrics may still be useful for background evaluation, but the live objective is closer to conversational adequacy under latency pressure.

7 A new benchmark: phrase completion and conversational adequacy

A Babelbit benchmark should evaluate source prefixes rather than only completed sentences. For each utterance, validators reveal prefixes incrementally, ask miners for one or more candidate outputs, and score each candidate against the full utterance and target interaction.

A benchmark item may contain:

1. Source audio or text stream.
2. Full source transcript for validation.
3. Optional reference translations.
4. Domain and register metadata.
5. Safety and terminology constraints.
6. Human or LLM-judged adequacy labels at each prefix.

The benchmark should report not merely final quality, but the frontier between quality and latency. A miner that emits excellent output at the end of the utterance is useful, but it has not solved the simultaneous problem. A miner that emits early but often wrongly is unsafe. The best miners push the Pareto frontier: earlier adequacy at the same risk, or lower risk at the same latency.

For example, in a French-English meeting benchmark, the input may be:

> *Je pense que vous avez tout a fait raison.*

A conventional output may be:

> *I think you are absolutely right.*

A Babelbit output may be:

> *Agreed.*

The benchmark should recognise that the second output may be superior when scored for timely conversational function.

8 Speech-mode transformer architecture

The transformer core remains the natural starting point because self-attention models long-range dependencies and supports parallel hypothesis construction. In its simplest form, attention maps queries, keys, and values according to

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V. \tag{16}$$

For streaming translation, the architecture must be constrained causally or chunkwise so that the system cannot depend on future source audio that has not arrived. The model must also maintain target-side continuity while source-side context grows.

A speech-mode transformer differs from a text-only LLM in at least four ways:

1. **Speech representation**: input may be continuous acoustic features, semantic speech tokens, acoustic tokens, or hybrid representations.
2. **Multistream structure**: source speech, target text, and target speech tokens may be processed synchronously.
3. **Incremental policy**: the model must decide when to emit target material under latency constraints.
4. **Vocoder integration**: target speech tokens must be rendered into audio without adding unacceptable delay.

The attraction of a one-shot process is that prediction, paraphrasing, translation, and speech generation can be learned jointly rather than stitched together as separate subsystems. This may reduce latency and avoid information loss at the ASR/text/TTS boundaries.

The design can still preserve a two-stream product architecture. A **low-latency stream** emits speech as soon as conversational adequacy is reached. A **high-accuracy stream** produces a translation of record with full context and no predictive shortcuts. The live listener receives smooth interaction; the meeting archive receives a trustworthy record.

9 Why Bittensor is the right development environment

Bittensor subnets are incentive-defined markets for digital commodities. Miners perform the useful work specified by the subnet; validators measure miner performance; emissions are allocated according to validator weights and Yuma Consensus. This fits Babelbit because the core research problem is open-ended, measurable, and naturally iterative.

A single central team can build a prototype. It is much harder for a small team to explore the full space of prediction policies, tokenisers, speech models, calibrators, domain adaptation strategies, summarisation objectives, safety judges, and deployment optimisations. A subnet can run many competitions in parallel or sequence, each with clear validation data and metrics. The system can reward improvements that reduce latency without harming adequacy, improve adequacy without increasing latency, reduce compute cost, add languages, or improve domain-specific performance.

Bittensor is especially appropriate because Babelbit's advantage is cumulative. Many subnets perform tasks whose outputs are consumed immediately. Babelbit's mining outputs can improve a production system over time: better models, better policies, better calibration data, better validation tools, and better customer-specific lexicons. The subnet therefore acts as a decentralised research and development engine.

The planned challenge sequence includes:

1. **Aggressive prediction**: general utterance completion and early adequacy.
2. **Vertical prediction**: medical, legal, industrial, education, government, and customer-support domains.
3. **Customer-specific lexicons**: EU, UN, corporate, and regulated-sector vocabularies.
4. **Summarisation and brevity**: shorter target outputs that preserve communicative function.
5. **Politeness and cultural adaptation**: safe, appropriate, locale-aware output.
6. **Clarification**: deciding when to ask, delay, or output a filler rather than guess.
7. **New languages**: adding language pairs with different reordering, morphology, and data availability.
8. **Speech-mode modelling**: replacing text bottlenecks with direct source and target speech representations.

Where Bittensor supports multiple incentive mechanisms, separate criteria can be weighted explicitly. Even with a smaller mechanism set, validators can use composite scores with auditable submetrics.

10 Competitive differentiation

Babelbit's first differentiating factor is conceptual. Large customer-facing translation systems still tend to define simultaneous translation in terms of speed and literal accuracy. That is not what professional interpreters do. Babelbit defines the gold standard as timely, adequate, context-sensitive communication.

The second differentiating factor is data and evaluation. Phrase-level adequacy labels, early-commit traces, calibrated confidence histories, and domain-specific conversational benchmarks are not generic MT assets. They are product-specific training infrastructure.

The third differentiating factor is the subnet itself. A Bittensor competition can run repeated optimisation rounds across all the criteria above. This permits a small organisation to mobilise a global pool of miners around tasks that would otherwise require a large internal laboratory.

The fourth differentiating factor is customer adaptation. Legal, medical, industrial, diplomatic, and enterprise contexts do not merely require different words. They require different thresholds for risk, politeness, compression, and clarification. Babelbit can treat those differences as separate mining and validation tasks.

11 Risks and mitigations

The central risk is harmful early commitment. A system that predicts aggressively can be wrong aggressively. The mitigation is not to avoid prediction, but to calibrate it, constrain it by context, and provide a high-accuracy stream of record.

A second risk is benchmark gaming. If miners can overfit to a public judge, they may optimise artefacts rather than user value. Babelbit should combine public metrics with held-out evaluation sets, private or obfuscated judge prompts, human anchor sets, pairwise preference checks, and versioned judge calibration.

A third risk is cultural overreach. Politeness and safety transformations must be configurable and auditable. Some contexts require literal preservation of offensive language; others require sanitisation. The product should make these policies explicit rather than hiding them inside the model.

A fourth risk is speech-token representation. Discrete tokens are attractive because they make speech compatible with language modelling, but continuous features can outperform discrete tokens in some spoken-language understanding settings. The subnet should not prematurely fix a single representation. It should allow miners to compete over tokenisation, feature design, and hybrid approaches.

12 Research roadmap

The subnet was launched with a competition aimed at validating early utterance completion in text. We are now going into Phase 2, which is to take our MVP and improve its performance, before going on to extending its feature set.

The roadmap can be summarised as follows:

Phase 1 — LLM-based utterance completion
Miner task: Complete utterances from sequential partial prefixes. *Validator score:* EATP, Lead, EA_γ, semantic adequacy.
Phase 2 — Speech-to-Speech translation from French to Englis
Miner task: Translate speech in real-time, with mandatory thresholds for accuracy and output speed, competing to improve relative latency. *Validator score:* Quality threshold, phrase completion latency.
Phase 3 — Extend to multiple languages and verticals]
Miner task: Some of the provided models are intrinsically multi-lingual, but they need to be trained or fine-tuned to get to the same performance level as French to English. *Validator score:* Mandatory thresholds for latency, with LLM-judge measuring accuracy in target languages.
Phase 4 — Basic Paraphrasing
Miner task: Remove interjections, repetitions; clarify meaning; shorten. *Validator score:* LLM-judge comparing literal translation of input with output.
Phase 5 — Nuanced Paraphrasing
Miner task: Remove expletives and offensive terms; reword to allow for typical cultural sensitivities. *Validator score:* End-user utility across domains.

13 Conclusion

Babelbit reframes real-time translation as an early adequacy problem rather than a faster literal translation problem. The key product question is: when does the system know enough to deliver useful meaning? Recent speech-mode transformer networks, discrete and continuous speech representations, streaming translation models, and multistream architectures make this question newly tractable. Professional interpreters show the desired behaviour: predict where safe, paraphrase for brevity, preserve the social act, and adapt to culture and context. Babelbit's contribution is to make that behaviour the objective of a Bittensor subnet.

By moving from text-only utterance completion to speech-to-speech translation, the subnet can progressively reward prediction, summarisation, politeness, clarification, domain adaptation, and new languages. This is precisely the kind of open-ended, measurable, iterative research problem that Bittensor is designed to accelerate. The result is not merely a translation engine; it is a human-centred communication layer for multilingual conversation.

A Appendices

A.1 LLM-as-judge protocol

An ongoing iterative work-stream by the core Babelbit team is to refine the ways in which LLMs can be used to assess the output of a *machine interpreter* on its own terms, rather than using the legacy approaches which apply to a *machine translator*. The Babelbit adequacy judge can operate reference-free or reference-based. In reference-free mode, the judge receives the source utterance and candidate output, plus metadata such as domain, register, and target locale. In reference-based mode, it also receives a trusted translation and is instructed to ignore superficial paraphrase.

The judge can return a category and calibrated score, for example:

Category	Meaning	Anchor
A	Fully adequate	1.00
B	Minor omission or acceptable paraphrase	0.85
C	Risky, incomplete, or pragmatically weak	0.55
D	Wrong, misleading, or off-topic	0.10
Abstain	Judge uncertain	conservative fallback

A monotone calibration function, such as isotonic regression, maps raw judge scores to empirical human adequacy. Pairwise preference training using Bradley–Terry or TrueSkill-style models can be used where relative judgments are more stable than absolute scores.

This work-stream is likely to become extremely complex, once the full range of paraphrasing objectives have been included, and there is a strong likelihood that this model will also benefit from being trained within the Bittensor ecosystem.

A.2 Reference list

References

[1] Ashish Vaswani et al. *Attention Is All You Need*. NeurIPS, 2017.

[2] Maja Popovic. *chrF: character n-gram F-score for automatic MT evaluation*. WMT, 2015.

[3] Tianyi Zhang et al. *BERTScore: Evaluating Text Generation with BERT*. ICLR, 2020.

[4] Ricardo Rei et al. *COMET: A Neural Framework for MT Evaluation*. EMNLP, 2020; subsequent COMET work through 2022.

[5] SEAMLESS Communication Team. *Joint speech and text machine translation for up to 100 languages*. Nature 637, 587–593, 2025. DOI: 10.1038/s41586-024-08359-z.

[6] SEAMLESS Communication Team. *Seamless: Multilingual Expressive and Streaming Speech Translation*. Meta AI Research, 2023.

[7] Tom Labiausse, Laurent Mazare, Edouard Grave, Patrick Perez, Alexandre Defossez, and Neil Zeghidour. *High-Fidelity Simultaneous Speech-To-Speech Translation*. arXiv:2502.03382, 2025.

[8] Keqi Deng, Wenxi Chen, Xie Chen, and Philip C. Woodland. *SimulS2S-LLM: Unlocking Simultaneous Inference of Speech LLMs for Speech-to-Speech Translation*. ACL 2025, pp. 16718–16734. DOI: 10.18653/v1/2025.acl-long.817.

[9] Siqi Ouyang, Xi Xu, and Lei Li. *InfiniSST: Simultaneous Translation of Unbounded Speech with Large Language Model*. arXiv:2503.02969, 2025.

[10] Siqi Ouyang, Xi Xu, and Lei Li. *CMU's IWSLT 2025 Simultaneous Speech Translation System*. arXiv:2506.13143, 2025.

[11] Pooneh Mousavi et al. *Discrete Audio Tokens: More Than a Survey!* arXiv:2506.10274, 2025.

[12] He Bai, Tatiana Likhomanenko, Ruixiang Zhang, Zijin Gu, Zakaria Aldeneh, and Navdeep Jaitly. *dMel: Speech Tokenization Made Simple*. Apple Machine Learning Research, 2025.

[13] Dingdong Wang, Junan Li, Mingyu Cui, Dongchao Yang, Xueyuan Chen, and Helen M. Meng. *Speech Discrete Tokens or Continuous Features? A Comparative Analysis for Spoken Language Understanding in SpeechLLMs*. EMNLP 2025.

[14] Bittensor Documentation. *Understanding Subnets*; *Yuma Consensus*; *Mining in Bittensor*; *Validating in Bittensor*. Accessed 2026.

[15] Babelbit. *Babelbit Subnet: source code, challenge specifications, and documentation*. https://github.com/babelbit. Accessed 2026.