QoS-QoE Translation with Large Language Model

Yingjie Yu
yyu69@illinois.edu
University of Illinois
Urbana-Champaign
Urbana, Illinois, USA

Mingyuan Wu
mw34@illinois.edu
University of Illinois
Urbana-Champaign
Urbana, Illinois, USA

Ahmadreza Eslaminia
ae15@illinois.edu
University of Illinois
Urbana-Champaign
Urbana, Illinois, USA

Lingzhi Zhao
lz26@illinois.edu
University of Illinois
Urbana-Champaign
Urbana, Illinois, USA

Kaizhuo Yan
kaizhuo2@illinois.edu
University of Illinois
Urbana-Champaign
Urbana, Illinois, USA

Klara Nahrstedt
klara@illinois.edu
University of Illinois
Urbana-Champaign
Urbana, Illinois, USA

A Appendix

A.1 Supervised Fine-tuning Instances

Figure 1 shows abridged chat-style supervised fine-tuning (SFT) instances for the two translation directions. Each instance is constructed from a single source-grounded relationship record. A system message instructs the model to output only the target field in valid JSON format. The user message contains the structured task input, including the task direction, contextual metadata, scenario, parameter mapping and definitions, source-grounded evidence, a partial history log, and a query. To create each training instance, we remove one time point from the original `history_log`. The observed side of that removed record is used as the `query`, and the corresponding target side is used as the assistant output. In the QoS→QoE setting, the query contains QoS value(s) and the assistant predicts the corresponding QoE value(s). In the QoE→QoS setting, the query contains QoE value(s) and the assistant predicts the corresponding QoS value(s).

A.2 Model Evaluation Details

For continuous-value prediction, we use MAPE and Accuracy@δ. This section provides the parameter-specific tolerance configuration for Accuracy@δ. We set the tolerance mode and δ value for each parameter based on its scale, unit, and semantic interpretation in the QoS-QoE translation setting. Because QoS and QoE parameters differ substantially in scale, units, and meaning, we use a parameter-specific δ rather than a single shared threshold. Table 1 reports the canonical parameter names, general definitions, and corresponding tolerance settings used in our evaluation. Notably, these general definitions are intended to provide a unified interpretation for evaluation, while the exact definition of the same parameter may vary slightly across different source papers.

Relative tolerances. Relative tolerances are used for scale-dependent rate-like quantities whose prediction error is more naturally interpreted proportionally than by a fixed absolute difference [11]. For `bitrate`, prior work also evaluates prediction quality using relative bitrate error, reflecting the fact that the same absolute error can correspond to very different levels of deviation across low and high bitrate ranges [10]. We use $\delta = 0.1$ for `bitrate`, which treats predictions within 10% of the ground truth as correct under Accuracy@δ. This provides a moderate proportional tolerance for near-correct predictions while preserving separation between materially different bitrate values. We use the same tolerance mode for `bitrate_change`, with $\delta = 0.15$.

Absolute tolerances. Absolute tolerances are used when prediction error is more naturally interpreted on the original scale, making deviations directly meaningful in the native unit or rating scale [3, 11]. In our setup, we use this tolerance mode for bounded

Table 1: Canonical parameter-specific tolerance configuration used for Accuracy@δ for model evaluation. Here, *rel* and *abs* denote relative and absolute tolerance, respectively.

Parameter	General definition	Mode	δ
bitrate	transmission rate	rel	0.1
qoe_score	generic QoE score	abs	0.3
mos	subjective opinion score	abs	0.3
vmaf	video quality metric	abs	3.0
rebuffering_time	stall duration	abs	0.5
jitter	packet delay variation	abs	5.0
packet_loss_rate	packet loss ratio	abs	0.01
latency	end-to-end delay	abs	10.0
rebuffering_ratio	stall ratio	abs	0.03
bitrate_change	bitrate switching magnitude	rel	0.15
rebuffer_count	stall count	abs	1.0
psnr	peak signal fidelity	abs	1.5
visqol	perceptual quality metric	abs	0.3
ssim	structural similarity	abs	0.02
vifp	visual information fidelity	abs	0.03
vqm	video quality metric	abs	0.05
pesq	perceptual speech quality	abs	0.25
peaq_odg	audio difference grade	abs	0.2
playback_rate	playback speed	abs	0.05
frame_rate	frames per second	abs	2.0
snr	signal-to-noise ratio	abs	2.0
ssim_db	SSIM in decibels	abs	1.5
total_rebuffering_time	cumulative stall duration	abs	1.0
segment_duration	segment duration	abs	0.25
video_delay_p95_ms	p95 video delay	abs	20.0
acr_grade	absolute category rating	abs	0.5

(a) QoS→QoE SFT instance.

(b) QoE→QoS SFT instance.

Figure 1: Abridged chat-style SFT instances for bidirectional QoS-QoE translation. Long text fields and repeated entries are abbreviated with "..." for readability.

quality scores, objective quality metrics, time-based metrics, ratios, counts, and delay-related metrics.

Bounded subjective and perceptual quality scores are defined on fixed rating scales, so their prediction error is more naturally interpreted on the original scale than as proportional error [6, 7]. This category includes `qoe_score` ($\delta = 0.3$), `mos` ($\delta = 0.3$), `visqol` ($\delta = 0.3$), `pesq` ($\delta = 0.25$), `peaq_odg` ($\delta = 0.2$), and `acr_grade` ($\delta = 0.5$). We set these δ values as small absolute tolerances on their scales to allow minor deviations while preserving meaningful quality differences.

Objective fidelity metrics are commonly evaluated on their own metric-specific scales [9, 12]. This category includes `vmaf` ($\delta = 3.0$), `psnr` ($\delta = 1.5$), `ssim` ($\delta = 0.02$), `vifp` ($\delta = 0.03$), and `vqm` ($\delta = 0.05$). We set these δ values as small absolute tolerances relative to the range and resolution of each metric.

Signal and native-unit rate metrics are expressed in engineering units such as decibels or frames per second [8, 11]. This category includes `snr` ($\delta = 2.0$), `ssim_db` ($\delta = 1.5$), and `frame_rate` ($\delta = 2.0$). We set these δ values as small absolute tolerances in their native units, allowing modest deviations while preserving materially different signal or temporal characteristics.

Time-based and delay-related metrics are measured in units such as seconds or milliseconds [2, 11]. This category includes `rebuffering_time` ($\delta = 0.5$), `total_rebuffering_time` ($\delta = 1.0$), `latency` ($\delta = 10.0$), `jitter` ($\delta = 5.0$), `segment_duration` ($\delta = 0.25$), `video_delay_p95_ms` ($\delta = 20.0$), and `playback_rate` ($\delta = 0.05$). We set these δ values as small absolute tolerances in their native units to allow near-correct timing predictions without conflating materially different delay behavior.

Ratio- and count-based parameters are interpreted on their original scales because they represent bounded fractions or discrete events rather than continuously scaled magnitudes [4, 11]. This category includes `packet_loss_rate` ($\delta = 0.01$), `rebuffering_ratio` ($\delta = 0.03$), and `rebuffer_count` ($\delta = 1.0$). We set these δ values as small absolute tolerances, allowing minor deviations for bounded ratios and a one-event tolerance for counts.

Table 2: Average rating and average confidence of each reviewer in the multi-reviewer evaluation stage.

LLM Reviewers	Avg. Rating	Avg. Confidence
Gemini-2.5-flash-lite	8.48	5.00
Claude-haiku-4-5-20251001	7.24	4.08
Grok-4.20-0309-reasoning	6.20	3.98

A.3 Multi-LLM reviewer Evaluation Details

To improve data quality, each metadata-enriched record is evaluated by three LLM reviewers: Gemini-2.5-flash-lite [5], Claude-haiku-4-5-20251001 [1], and Grok-4.20-0309-reasoning [13]. Each reviewer returns a rating, a confidence score, and textual feedback. Table 2 summarizes the average rating and average confidence of each reviewer on the first-round LLM review, providing a coarse view of reviewer behavior in our pipeline. The results suggest that the initial extraction quality is already reasonably strong before re-evaluation. In our setup, Gemini-2.5-flash-lite gives the highest average rating and confidence, while Grok-4.20-0309-reasoning is comparatively more conservative. Claude-haiku-4-5-20251001 lies between the other two reviewers in both average rating and confidence.

A.4 Pipeline Prompt Details

The full prompts used in our pipeline are provided in the released codebase.[1] In particular, `relationship_extraction.txt` is used for source-grounded QoS-QoE relationship extraction from parsed

[1] https://github.com/yyu6969/qos-qoe-translation/tree/main/src/prompts

papers, `metadata_enrichment.txt` is used for contextual metadata generation and normalization, and `data_evaluation.txt` is used in the multi-reviewer data evaluation stage. Together, these prompts define the task-specific instructions and structured output requirements used in the dataset construction pipeline.

References

[1] Anthropic. 2025. Introducing Claude Haiku 4.5. https://www.anthropic.com/news/claude-haiku-4-5 Official model announcement. Accessed: 2026-04-01.

[2] J. J. Aranda, M. Cortés, J. Salvachúa, M. Narganes, and I. Martínez-Sarriegui. 2020. *The Quality for Service (Q4S) Protocol.* RFC 8802. RFC Editor. https://doi.org/10.17487/RFC8802

[3] Silvia Beddar-Wiesing, Alice Moallemy-Oureh, Marie Kempkes, and Josephine M. Thomas. 2025. Absolute Evaluation Measures for Machine Learning: A Survey. https://doi.org/10.48550/arXiv.2507.03392 arXiv:2507.03392 [cs.LG]

[4] Florin Dobrian, Vyas Sekar, Asad Awan, Ion Stoica, Dilip Joseph, Aditya Ganjam, Jibin Zhan, and Hui Zhang. 2011. Understanding the Impact of Video Quality on User Engagement. *ACM SIGCOMM Computer Communication Review* 41, 4 (2011), 362–373. https://doi.org/10.1145/2043164.2018478

[5] Google. 2025. Gemini 2.5 Flash-Lite. https://ai.google.dev/gemini-api/docs/models/gemini-2.5-flash-lite Official model documentation. Accessed: 2026-04-01.

[6] International Telecommunication Union. 1996. *Methods for Subjective Determination of Transmission Quality.* Recommendation P.800. ITU-T. https://www.itu.int/rec/T-REC-P.800-199608-I

[7] International Telecommunication Union. 2005. *Application Guide for Objective Quality Measurement Based on Recommendations P.862, P.862.1 and P.862.2.* Recommendation P.862.3. ITU-T. https://www.itu.int/rec/T-REC-P.862.3-200511-S

[8] Alex MacKin, Aaron Zhang, and David R. Bull. 2015. A Study of Subjective Video Quality at Various Frame Rates. In *2015 IEEE International Conference on Image Processing (ICIP).* 3407–3411. https://doi.org/10.1109/ICIP.2015.7351436

[9] Margaret H. Pinson and Stephen Wolf. 2004. A New Standardized Method for Objectively Measuring Video Quality. *IEEE Transactions on Broadcasting* 50, 3 (2004), 312–322. https://doi.org/10.1109/TBC.2004.834028

[10] Taveesh Sharma, Tarun Mangla, Arpit Gupta, Junchen Jiang, and Nick Feamster. 2023. Estimating WebRTC Video QoE Metrics Without Using Application Headers. In *Proceedings of the 2023 ACM Internet Measurement Conference (IMC '23).* Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3618257.3624828

[11] Juan Terven, Diana-Margarita Cordova-Esparza, Julio-Alejandro Romero-González, Alfonso Ramírez-Pedraza, and E. A. Chávez-Urbiola. 2025. A comprehensive survey of loss functions and metrics in deep learning. *Artificial Intelligence Review* 58, Article 195 (2025). https://doi.org/10.1007/s10462-025-11198-7

[12] Abhinau K. Venkataramanan, Cosmin Stejerean, Ioannis Katsavounidis, and Alan C. Bovik. 2024. One Transform to Compute Them All: Efficient Fusion-Based Full-Reference Video Quality Assessment. *IEEE Transactions on Image Processing* 33 (2024), 509–524. https://doi.org/10.1109/TIP.2023.3345227

[13] xAI. [n. d.]. Grok 4.20 0309 Reasoning. https://docs.x.ai/developers/models/grok-4.20-0309-reasoning Official model documentation. Accessed: 2026-04-01.