

TurnStyle: Suggesting Style Prompts for Instruction-Guided Text-to-Speech Using Bimodal Dialogue Contexts

Jack McLaughlin

Department of Computer Science
Rochester Institute of Technology
Rochester, NY 14586
jdm8943@rit.edu

Cecilia O. Alm

School of Psychology and Cognitive Science
Rochester Institute of Technology
Rochester, NY 14586
coagla@rit.edu

Abstract

Instruction-guided text-to-speech (TTS) systems are among the most intuitive for controllable speech synthesis, but writing descriptive instructions in the form of style prompts is too cumbersome for users to maintain natural conversation rhythm. We introduce TurnStyle, a prompt generation system, that produces style prompts for instruction-guided TTS, leveraging a bimodal dialogue history (speech and text transcripts) and a user script query. To address the unintuitive interface problem, TurnStyle creates a style prompt by conditioning the Llama-3.2-3B-Instruct model on a prefix vector generated from an embedded bimodal representation of a dialogue. Computational experiments with performance measured across standard metrics like BERTscore and ROUGE-L show that TurnStyle’s architecture improved on a comparative baseline model for the understudied task of style prompt generation. The introduced model also consumed over 15 times less energy than the baseline and required less memory, suggesting that the TurnStyle framework may feasibly run on edge devices.

1 Introduction

An intuitive text-to-speech (TTS) user interface is required for day-to-day conversational or clinical applications, like vocal prostheses, so that users can more quickly produce expressive speech and maintain natural-sounding conversation. While current commercial expressive speech synthesis (ESS) interfaces, such as ElevenLabs and Speechify, provide expressive controls, they are not intuitive (Brade et al., 2025). This work is partially motivated by the present lag in ESS tools to provide intuitive controls for users.

One solution for intuitive, conversational TTS is instruction-guided TTS systems. These models used a natural-language style prompt to control the expressivity of synthesized speech. However, Brade et al. (2025) found, using a Wizard-of-Oz

study, that users preferred when the system generated style prompts for them. This work proposes TurnStyle, a framework that seeks to generate a context-aware style prompt that would be used as input for an instruction-guided TTS system to synthesize the user’s desired speech.

TurnStyle combines and expands the methodology of Zhao et al. (2023) and Yamauchi et al. (2024) to address the understudied style prompt generation task. Reimplementing the description of a dialogue encoder architecture by Zhao et al. (2023), and adding a dialogue pooling step, TurnStyle generates an embedded representation of the bimodal (speech and text transcripts) dialogue context. Leveraging StyleCap’s architecture (Yamauchi et al., 2024), the dialogue embedding is processed with a transformer stack to produce a prefix vector, to condition Llama-3.2-3B-Instruct which generates the style prompt.

Through experimentation, this work tests the hypothesis that an accurate style prompt can be predicted from the current script, as well as both the transcript and speech clips from the conversation context. To train and test TurnStyle, two datasets were used, including ParaSpeechCaps (Diwan et al., 2025), specifically the Espresso subset (Nguyen et al., 2023), and StyleTalk (Lin et al., 2024). Next, to evaluate style prompt quality, BERTscore (Zhang et al., 2020), METEOR score (Banerjee and Lavie, 2005), ROUGE-L (Lin, 2004), and chrF++ (Popović, 2017), were used to compare the generated prompts against the gold standard style descriptions of the test sets. Results additionally include F1-score for distinct keyword categories. As a comparative baseline, results include scores for style prompts created by out-of-the-box Llama-3.2-3B-Instruct performing few-shot generation. The results, presented in Section 4, showed that TurnStyle improves on the style prompt generation task over the baseline, across all evaluation metrics, while being more resource-efficient.

2 Related Work

2.1 Instruction Guided Text-to-Speech

Instruction-guided TTS leverages natural language style prompts to control speech synthesis. Guo et al. (2023) proposed a method that encodes style prompts into BERT embeddings, which condition both the content encoder and the speech decoder. Sigurgeirsson and King (2024) proposed an automated version of instruction-guided TTS, which prompted an LLM to directly modify the F_0 , duration, and energy values of phoneme units before they are decoded into a mel-spectrogram output. Yang et al. (2024) employed cross-modal learning strategies to align speech and text style embeddings into the same latent space. Lyth and King (2024) integrated content and style prompts via cross-attention and represented them using the Descript Audio Codec. TurnStyle complements these systems by generating style prompts for the user automatically, based on dialogue history. The model proposed by Jeon et al. (2025) learned a similar task to TurnStyle. Their system generated intermediate style labels for a target utterance by prompting an LLM with the target utterance and a transcription of the preceding conversation history. Rather than single-word style labels, TurnStyle generates full-sentence style descriptions and incorporates speech as input during dialogue modeling.

2.2 Modeling Dialogue History

Modeling dialogue context is essential for expressive and contextually appropriate speech synthesis, and by extension, style prompt generation. Cong et al. (2021) used the previous utterance’s transcription as context for their style embedding, while Hu et al. (2024) encoded dialogue context at the word- and utterance-level to capture hierarchical information. Fathima et al. (2025) proposed a multi-scale style extractor that learned global speaker embeddings, sentence-level embeddings for contextual prosody, and subword embeddings for phoneme-level variations. TurnStyle generates an embedding from the entire bimodal dialogue, which is transformed into a natural language style prompt by an LLM; thus offering a user the chance to edit the style prompt before it is used to synthesize expressive speech using an instruction-guided TTS model. Zhao et al. (2023) embedded inputs at both the utterance and global conversation level, and in both the speech and text modalities to capture speaker-dependent temporal relationships for the

emotion-recognition task. TurnStyle extends their architecture with a dialogue-pooling step to get one embedding for the whole dialogue, rather than one embedding per utterance.

Additionally, this work builds upon StyleCap, a method proposed by Yamauchi et al. (2024). Their model learned a prefix vector, or soft prompt, from a speech clip, which an LLM used to generate a style description for that speech clip. To extend their work to style prompt *predictions*, rather than style *captions*, TurnStyle encodes an entire dialogue as a prefix vector.

3 Methodology

3.1 Task Formalization and Architecture

TurnStyle combines StyleCap, the prefix vector generation strategy proposed by Yamauchi et al. (2024), and a modified dialogue-encoder from the emotion classifier architecture of Zhao et al. (2023), to move toward context-appropriate style prompt generation. In essence, TurnStyle predicts the next utterance’s style description from speech clips and transcripts of the preceding conversation.

A dialogue is represented as a chronologically ordered sequence of T turns that is a slice of a larger conversation from the data. Let the t -th turn be the tuple $u_t = (w_t, s_t, x_t)$, with $t = 1, \dots, T$. Moreover, w_t is the waveform of utterance t , while s_t is the speaker identity, from the set $\mathcal{S}$ of speaker labels, and x_t is the text transcription of utterance t . During training, the final turn of input differs because it is treated as text-only: the speech of the final turn is zeroed out. During inference, there is no speech of the final turn since it has yet to be synthesized, so it should not be seen during training.

Figure 1 details the TurnStyle architecture. Formally, the model learns the mapping $f_\theta : \mathcal{C} \mapsto \hat{y}$, where $\mathcal{C} = (u_1, \dots, u_T)$ is a bimodal dialogue, $u_T = (0, s_T, x_T)$ and $\hat{y}$ is the predicted natural-language description of the speaking style for synthesizing w_T . The style description of the final turn of dialogue u_T , is y , which serves as the ground-truth to compare $\hat{y}$ against during training.

3.2 Dialogue Encoder

The proposed dialogue-modeling approach was inspired by and sought to reimplement Zhao et al. (2023), while also including an additional self-attentive pooling step over the utterance dimension to produce the final dialogue embedding. In the

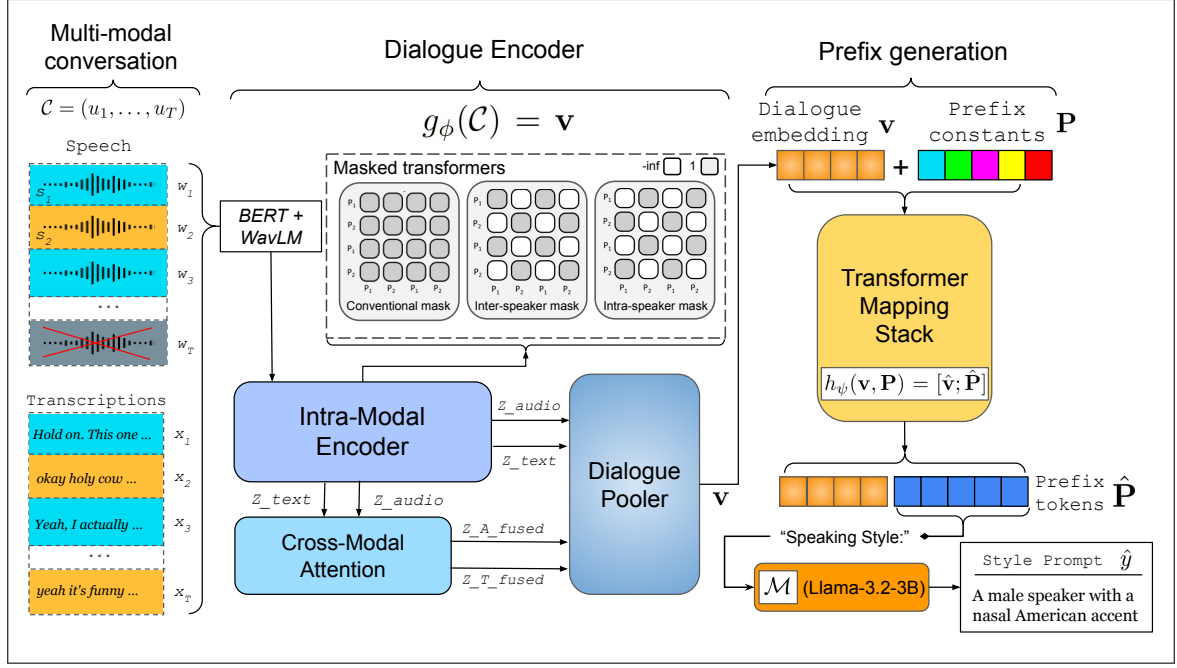


Figure 1: The TurnStyle model architecture. The system decomposes as $f_\theta = \mathcal{M} \circ h_\psi \circ g_\phi$, where $g_\phi(\mathcal{C}) = \mathbf{v}$ is a learned dialogue encoding, h_ψ maps $\mathbf{v}$ and trainable embeddings $\mathbf{P}$ to $[\hat{\mathbf{v}}; \hat{\mathbf{P}}]$ jointly, and $\mathcal{M}$ is a frozen autoregressive language model that generates $\hat{y}$ conditioned on the prefix $\hat{\mathbf{P}}$. Concretely, $h_\psi(\mathbf{v}, \mathbf{P}) = [\hat{\mathbf{v}}; \hat{\mathbf{P}}]$, where $\mathbf{v}$ is discarded after the forward pass and $\hat{\mathbf{P}}$ serves as the soft prompt.

experiments, the text and speech modalities of utterances were embedded with BERT-base (Devlin et al., 2019) and WavLM-Base-plus (Chen et al., 2022). Turn-positions were then encoded with a learned lookup table.

3.2.1 Intra-Modal Dialogue Encoder and Cross-Modal Attention

Each modality was processed independently at this stage, following Zhao et al. (2023). Inspired by their description, the Intra-Modal Dialogue Encoder component consisted of three transformer-based subcomponents (conventional causal attention, as well as *intra*- and *inter*-speaker aware attention), which intended to model the influence of interlocutors’ styles on each other during the conversation. The *intra*-speaker transformer combined a causal mask and a cross-speaker mask, so during self-attention, utterances attend to other utterances by the same speaker, while the *inter*-speaker transformer restricted attention to utterances by other speakers and their influence on the current speaker. To produce two per-modality *intra*-modal dialogue embeddings, these outputs were concatenated, fused with the original utterance embedding through residual connections and layer normalization, and passed through a feed-forward network.

Next, the Cross-Modal Attention component, inspired again by Zhao et al. (2023), produced two *inter*-modal dialogue embeddings, one per modality. By calculating attention with queries of one modality and keys and values of the other modality, it intended to encode relevant style information from the opposing modality. The two components thus resulted in four embeddings.

3.2.2 Dialogue Pooling

TurnStyle adds a dialogue pooling component to the architecture, which differed from Zhao et al. (2023), whose design focused on individual utterances. In contrast, TurnStyle encodes all utterances into one embedding. For each utterance, the two *intra*-modal representations and the two *inter*-modal representations were concatenated. Self-attentive pooling was then applied across the utterance dimension to encode the entire dialogue into a single fixed-length vector representing its full context. Dialogue pooling thus intended to capture the bimodal dialogue context in one embedding for prefix vector generation.

3.3 Prefix Vector Generation

TurnStyle’s prefix generation architecture followed the approach of the StyleCap model (Yamauchi et al., 2024). To generate the prefix vector, the dia-

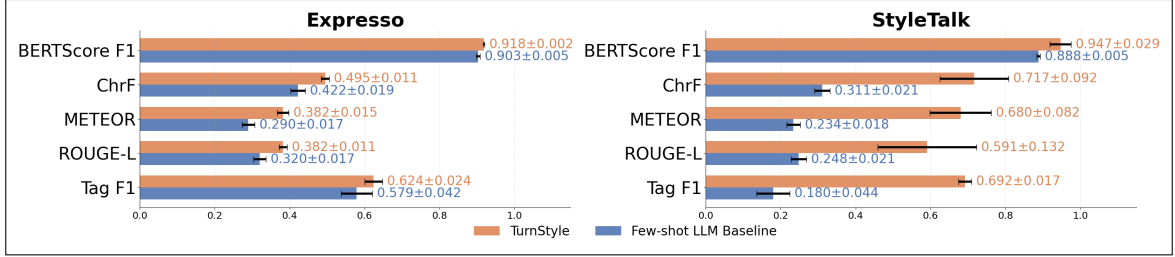


Figure 2: TurnStyle consistently outperformed the baseline, indicating the effectiveness of its dialogue modeling approach. BERTscore (Zhang et al., 2020), METEOR score (Banerjee and Lavie, 2005), ROUGE-L (Lin, 2004), and chrF++ metric (Popović, 2017) were used as semantic and lexical similarity metrics to compare predicted output with reference data. Tag F1-score is the aggregate F1-score for reproducing specific style tags from the input. See Table 1 in the appendix for the categories and values of tags. Created with Matplotlib (Hunter, 2007).

logue embedding and K learnable constants were processed with a stack of transformers, and the embedding was discarded, leaving the attended constants. These constants (the prefix vector) conditioned an LLM to generate the style prompt output, specifically with Llama-3.2-3B-Instruct.

3.4 Datasets

For computational experiments, training and testing used the Espresso (Nguyen et al., 2023) subset of the ParaSpeechCaps (Diwan et al., 2025) dataset, because of its conversational organization, as well as the StyleTalk dataset (Lin et al., 2024). The Espresso subset has 4 speakers and 23 hours of freeform dialogue. After pre-processing (see appendix section A.1) there were 2,311 total conversations of varying length. StyleTalk is a synthetically generated conversational speech dataset comprising 7 hours of speech and 2,235 conversations, each with 5 turns, where the first 3 turns are transcript only, while the next 2 turns include both transcript and speech; experimentation focused on the final 3 when extracting dialogues.

3.5 Experimental Setup

The training set was a mix of dialogues from both datasets, ParaSpeechCaps/Espresso and StyleTalk, and included 4,351 total dialogues. The order of dialogues was randomized between trials, but the within-dialogue turn order was retained. Dialogues with a length of 3 turns were considered, as decided from experimentation (see appendix Table 4). Test sets of 200 dialogues from each dataset were used in evaluation. While a given speaker could appear in both the train and test splits, the training and test dialogue instances were pulled from disjoint collections of conversations. Models were

trained with the AdamW optimizer, batch size of 16, for 15 epochs with early stopping. Training leveraged teacher-forced cross entropy loss. The transformer mapping network had 2 layers, and the soft-prompts were a prefix vector of size 10. These values were selected via experimentation (see appendix Table 4 for an outline, and Tables 5 through 7 for results). Other hyperparameters were chosen in a hyperparameter search (see appendix Table 3). The phrase "*Speaking Style:*" was added to both TurnStyle’s and the baseline’s input prompts to guide the generation.

As a comparative baseline, experimentation used Llama-3.2-3B-Instruct (Grattafiori et al., 2024) with few-shot inference. For each inference, this LLM saw 25 examples of style prompt generation, where each example was an input-output mapping of a transcript-only dialogue context and the gold-standard style description of the final turn of that dialogue, and was then instructed to predict a style prompt for a new dialogue context.

4 Experimental Results

Performance across various metrics for TurnStyle and the baseline is shown in Figure 2, showing that TurnStyle generally performs better than the baseline. The metric results on the Espresso test set are more consistent, likely due to the fact that Espresso had more speech hours and more complex reference annotations than StyleTalk, allowing the model to learn more consistent patterns. Nevertheless, TurnStyle and the baseline both scored around 0.9 on BERTScore F1, indicating that predictions are semantically similar enough to the gold standard comparison to result in no meaningful difference in score. This may be due to the short length and narrow language domain of style prompts.

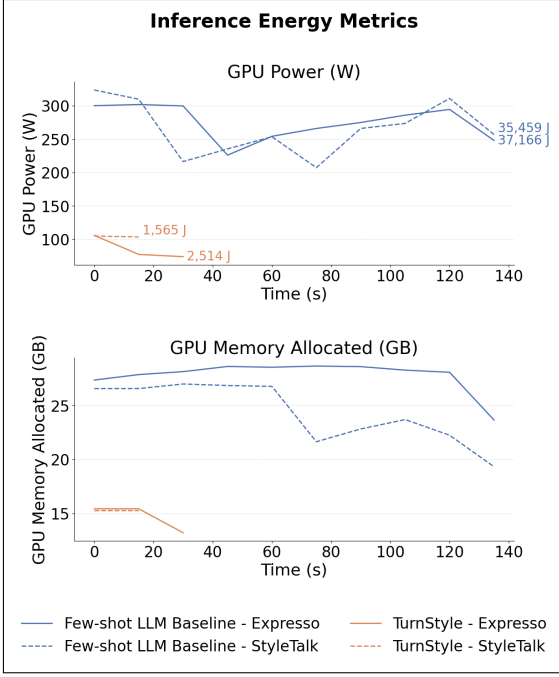


Figure 3: As shown, TurnStyle is more energy and memory efficient during inference. Also, running TurnStyle for inference drew 50-100 watts of power, while the LLM baseline drew around 250 watts. TurnStyle’s memory use peaked at 15 GB and the baseline peaked at around 27 GB. Additionally, TurnStyle inferred the test sets in around 30 seconds, while the LLM baseline took over 2 minutes.

4.1 Memory and Energy Efficiency

Figure 3 summarizes the inference efficiency results over all 200 test set instances with a batch size of 16, on an A100 GPU. The baseline used over $15\times$ more energy than TurnStyle, and it required around $1.5\times$ more memory at any given point during inference. The LLM component of TurnStyle only ever infers over 10-20 tokens at a time, from the prefix vector plus the system prompt “*Speaking Style:*”, while the baseline’s input is comprised of hundreds of tokens from the 25 few-shot examples of dialogues processed in each inference. Thus, TurnStyle is much more token efficient (and therefore faster, more energy efficient, and more memory efficient) for a higher-scoring output. This suggests TurnStyle might be feasible for edge-computing, which would preserve user-privacy as their un-encoded recorded speech would never leave their device. Future work should explore this applicability.

5 Conclusion

This work introduces TurnStyle, a system that automatically generates natural-language style prompts

from bimodal dialogue history. The experimental results indicate that the conversational context is useful for style prompt generation. Furthermore, the inference efficiency of TurnStyle suggests it may be feasible for use on edge devices, which would preserve user privacy in a real-world setting.

Limitations

Due to the limited number of real speakers in the data, further experimentation must be performed to validate that the model can generalize to real-world data and out-of-distribution scenarios. Moreover, further efforts are needed in data annotation of style prompts on conversational speech datasets having many speakers. Furthermore, both Espresso and StyleTalk skew toward American English speakers, meaning the model may generalize less well to speakers of other varieties of English and cultural communication norms. Additional work should also evaluate model confidence, since it would be important to signal to a user when the model is uncertain and offer controls to override its suggestions.

Additionally, metric choice is crucial when assessing performance on this task. Semantic metrics like BERTscore can be insufficient because style descriptions have similar semantic meaning even if important details like style tag keywords differ. Metrics like METEOR and ROUGE-L can also fall short because the generated style prompt could be valid, even if there is a word-level mismatch when compared to the gold standard. Moreover, a human-subject evaluation that tests TurnStyle’s usability would be important but is left to future work.

Ethical Considerations

One benefit of the TurnStyle framework is its energy efficiency. Energy efficient models can run on edge devices, avoiding a user’s data leaving their device, preserving privacy and protecting sensitive data like personal conversations. While TurnStyle is more efficient than the naive baseline considered in this work, more work would be required to verify its feasibility on edge devices.

Acknowledgement

The authors acknowledge Research Computing at the [Rochester Institute of Technology \(2026\)](#) for providing computational resources and support that have contributed to the research results reported in this work.

References

- Satanjeev Banerjee and Alon Lavie. 2005. [METEOR: An automatic metric for MT evaluation with improved correlation with human judgments](#). In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.
- Stephen Brade, Sam Anderson, Rithesh Kumar, Zeyu Jin, and Anh Truong. 2025. [SpeakEasy: Enhancing text-to-speech interactions for expressive content creation](#). In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, CHI '25, New York, New York, USA. Association for Computing Machinery.
- Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, Jian Wu, Long Zhou, Shuo Ren, Yanmin Qian, Yao Qian, Jian Wu, Michael Zeng, Xiangzhan Yu, and Furu Wei. 2022. [WavLM: Large-scale self-supervised pre-training for full stack speech processing](#). *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518.
- Jian Cong, Shan Yang, Na Hu, Guangzhi Li, Lei Xie, and Dan Su. 2021. [Controllable context-aware conversational speech synthesis](#). In *Proceedings of Interspeech 2021*, pages 4658–4662, Brno, Czechia. International Speech Communication Association.
- 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, Minneapolis, Minnesota, USA. Association for Computational Linguistics.
- Anuj Diwan, Zhisheng Zheng, David Harwath, and Eunso Choi. 2025. [Scaling rich style-prompted text-to-speech datasets](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 3639–3659, Suzhou, China. Association for Computational Linguistics.
- Rahmath Fathima, Shruthi Ramesh, Ansh Bomb, Rida Eshal Chuna, and S. Chitrakala. 2025. [ECHO-TTS: Expressive, contextual, and harmonized order text-to-speech \(TTS\) using deep learning techniques](#). In *Proceedings of 2025 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*, pages 1–6, Bengaluru, India.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. [The Llama 3 herd of models](#). *Preprint*, arXiv:2407.21783.
- Zhifang Guo, Yichong Leng, Yihan Wu, Sheng Zhao, and Xu Tan. 2023. [PromptTTS: Controllable text-to-speech with text descriptions](#). In *Proceedings of ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, Rhodes Island, Greece.
- Yifan Hu, Rui Liu, Guanglei Gao, and Haizhou Li. 2024. [FCTalker: Fine and coarse grained context modeling for expressive conversational speech synthesis](#). In *Proceedings of 2024 IEEE 14th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, pages 299–303, Beijing, China.
- John D. Hunter. 2007. [Matplotlib: A 2D graphics environment](#). *Computing in Science & Engineering*, 9(3):90–95.
- Yejin Jeon, Youngjae Kim, Jihyun Lee, and Gary Lee. 2025. [Prompt-guided selective masking loss for context-aware emotive text-to-speech](#). In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 638–650, Albuquerque, New Mexico. Association for Computational Linguistics.
- Chin-Yew Lin. 2004. [ROUGE: A package for automatic evaluation of summaries](#). In *Proceedings of Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.
- Guan-Ting Lin, Cheng-Han Chiang, and Hung-yi Lee. 2024. [Advancing large language models to capture varied speaking styles and respond properly in spoken conversations](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6626–6642, Bangkok, Thailand. Association for Computational Linguistics.
- Dan Lyth and Simon King. 2024. [Natural language guidance of high-fidelity text-to-speech with synthetic annotations](#). *Preprint*, arXiv:2402.01912.
- Ron Mokady, Amir Hertz, and Amit H. Bermano. 2021. [ClipCap: CLIP prefix for image captioning](#). *Preprint*, arXiv:2111.09734.
- Tu Anh Nguyen, Wei-Ning Hsu, Antony D’Avirro, Bowen Shi, Itai Gat, Maryam Fazel-Zarani, Tal Reiz, Jade Copet, Gabriel Synnaeve, Michael Hasid, Felix Kreuk, Yossi Adi, and Emmanuel Dupoux. 2023. [EXPRESSO: A benchmark and analysis of discrete expressive speech resynthesis](#). In *Proceedings of Interspeech 2023*, pages 4823–4827, Dublin, Ireland.
- Maja Popović. 2017. [chrF++: words helping character n-grams](#). In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark. Association for Computational Linguistics.
- Rochester Institute of Technology. 2026. [Research computing services](#).

Atli Sigurgeirsson and Simon King. 2024. [Controlable speaking styles using a large language model](#). In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 10851–10855, Seoul, Korea.

Wes McKinney. 2010. [Data structures for statistical computing in Python](#). In *Proceedings of the 9th Python in Science Conference*, pages 56 – 61, Austin, Texas, USA.

Kazuki Yamauchi, Yusuke Ijima, and Yuki Saito. 2024. [STYLECAP: Automatic speaking-style captioning from speech based on speech and language self-supervised learning models](#). In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 11261–11265.

Dongchao Yang, Songxiang Liu, Rongjie Huang, Chao Weng, and Helen Meng. 2024. [InstructTTS: Modelling expressive TTS in discrete latent space with natural language style prompt](#). *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:2913–2925.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. [BERTScore: Evaluating text generation with BERT](#). In *Proceedings of International Conference on Learning Representations*, Addis Ababa, Ethiopia.

Huan Zhao, Bo Li, and Zixing Zhang. 2023. [Speaker-aware cross-modal fusion architecture for conversational emotion recognition](#). In *Proceedings of Interspeech 2023*, pages 2718–2722, Dublin, Ireland.

A Dataset Preparation and Analysis

A.1 Preprocessing Pipeline

The audio for Espresso and StyleTalk datasets were downloaded using a custom script. Espresso was preprocessed by applying voice-activity detection (VAD) segmentation and loudness normalization, and annotations from ParaSpeechCaps (Diwan et al., 2025) for Espresso were downloaded. Through sentence-templating, StyleTalk’s categorical style labels were turned into sentences. The order of the keyword categories in the template was shuffled to generalize the model against the order of keywords. Using metadata from the filenames, a conversational index was created to organize the utterances in both datasets into contiguous conversations. Some utterances from Espresso had missing annotations, so to maintain contiguous conversations for training, these were split into separate conversations at the missing annotation.

A.2 Analysis

There are 59 tags, either situational or intrinsic, for the Espresso subset in the ParaSpeechCaps dataset (Diwan et al., 2025), shown in Figures 4 and 5. One trend within the situational tags is that vague descriptor words like *loud* and *animated* are over-represented and more precise words like *sleepy* are underrepresented, meaning predicted style prompts might tend towards generic style descriptors.

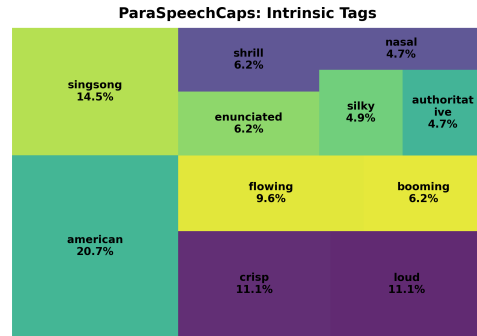


Figure 4: Diwan et al. (2025) describes intrinsic tags as inherent to the speaker’s way of talking. Squares’ areas represent their proportions. This figure was regenerated.

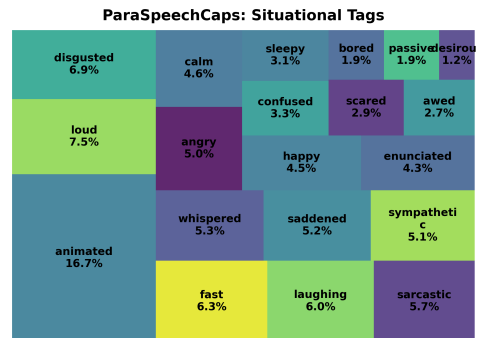


Figure 5: Diwan et al. (2025) describes situational tags as the utterance’s style. Squares’ areas represent their proportions. This figure was regenerated.

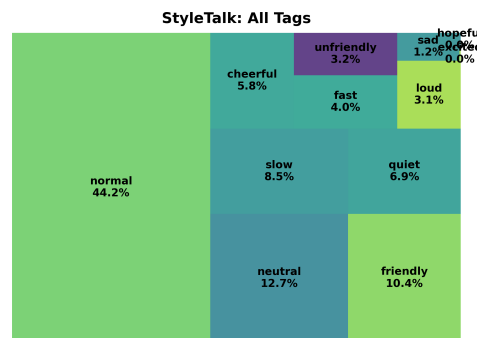


Figure 6: Style tags in StyleTalk (Lin et al., 2024) are less descriptive than those in ParaSpeechCaps:Espresso, possibly impacting model generalization.

Dataset	Category	Values
Expresso	Accent	american
	Gender	female · male
	Intrinsic Tags	authoritative · booming · crisp · enunciated · flowing · loud · nasal · shrill · silky · singsong
	Situational Tags	angry · animated · awed · bored · calm · confused · desirous · disgusted · enunciated · happy · laughing · loud · passive · saddened · sarcastic · scared · sleepy · sympathetic · whispered
	Noise	clean environment · noisy environment · environment balanced
	Pitch	high-pitched
	Speaking Rate	fast speed · measured speed · slow speed
StyleTalk	Emotion	cheerful tone · excited tone · friendly tone · hopeful tone · neutral tone · sad tone · unfriendly tone
	Speaking Rate	fast speed · normal speed · slow speed
	Volume	loud volume · normal volume · quiet volume

Table 1: Style tags for each tag category in ParaSpeechCaps:Expresso (Diwan et al., 2025) and StyleTalk (Lin et al., 2024). Values obtained using *pandas* (Wes McKinney, 2010).

Hyperparameter	Value
Epochs	15
Batch Size	16
Dialogue Length	3
Mapping Network	8 attention heads, 2 layers
Dialogue Encoder Dimension	768
Feedforward Dimension	2048
Dialogue Encoder Layers	2 context, 2 speaker
Prefix Vector Length	10
Max Audio Length	15s
Sample Rate	16,000 Hz
Seed	42

Table 2: These parameters were held constant during the hyperparameter search experiments in Table 3.

Hyperparameter	Range / Values	Search Type	Selected Value
Learning rate	[1e-5, 1e-3]	Log-uniform sampling	8e-5
Weight decay	[1e-4, 1e-2]	Log-uniform sampling	4e-3
Warmup ratio	[0.0, 0.2]	Uniform sampling	0.16
Dropout	{0.1, 0.2, 0.3}	Discrete values	0.1
Gradient clipping	{0.5, 1.0, 2.0}	Discrete values	1.0
LR schedule	{cosine, linear, constant}	Discrete values	linear

Table 3: Bayesian hyperparameter search performed over these parameter value ranges resulted in the value provided in the **Selected Value** column.

Experiment Condition	Values Tested	Notes
Dialogue Length	1, 3, 5 turns	We chose to cap the search at 5 since that is the maximum number of turns in any StyleTalk conversation, and $\sim 80\%$ of Espresso conversations are 5 turns or fewer.
Prefix Vector Length K	5, 10, 20 learnable tokens	Prior work (ClipCap (Mokady et al., 2021), StyleCap (Yamauchi et al., 2024)) tested this parameter as well and typically saw diminishing returns around $K=10$.
Mapping Network Depth	2, 4, 8 layers	8 layers follows prior work (ClipCap (Mokady et al., 2021), StyleCap (Yamauchi et al., 2024)), but this number of layers significantly bloats the total parameters without much improvement on performance (see Table 5).

Table 4: Three experiments were conducted for determining *Dialogue Length*, *Prefix Vector Length*, and *Mapping Network Depth*. Tables 5, 6, and 7 present comprehensive results.

Test Set	Mapping Network Depth	BERTScore F1	METEOR	ChrF	ROUGE-L	Tag F1	Inference Time (s)
expresso	2	0.918 ± 0.002	0.381 ± 0.014	0.494 ± 0.011	0.383 ± 0.011	0.622 ± 0.022	44.540 ± 6.061
	4	0.919 ± 0.001	0.390 ± 0.008	0.499 ± 0.006	0.383 ± 0.008	0.602 ± 0.024	50.466 ± 4.327
	8	0.917 ± 0.006	0.372 ± 0.045	0.488 ± 0.030	0.372 ± 0.031	0.533 ± 0.038	49.984 ± 5.864
	Baseline Naive LLM (25-shot, 5 turns ea.)	0.903 ± 0.004	0.295 ± 0.014	0.426 ± 0.015	0.324 ± 0.012	0.576 ± 0.027	168.609 ± 5.303
styletalk	2	0.944 ± 0.032	0.662 ± 0.094	0.705 ± 0.107	0.574 ± 0.150	0.688 ± 0.013	19.741 ± 5.819
	4	0.935 ± 0.027	0.651 ± 0.067	0.688 ± 0.078	0.567 ± 0.106	0.650 ± 0.049	20.839 ± 4.164
	8	0.927 ± 0.044	0.489 ± 0.269	0.534 ± 0.260	0.444 ± 0.259	0.355 ± 0.324	20.532 ± 4.633
	Baseline Naive LLM (25-shot, 5 turns ea.)	0.889 ± 0.003	0.238 ± 0.015	0.312 ± 0.018	0.251 ± 0.015	0.179 ± 0.040	161.750 ± 5.047

Table 5: Model performance varying the *Mapping Network Depth* of the transformer stack which performs prefix vector generation, compared to the few-shot generation LLM baseline for reference. The standard deviation across 5 trials is reported. The value 2 was chosen for use in the final model.

Test Set	Prefix Vector Length	BERTScore F1	METEOR	ChrF	ROUGE-L	Tag F1	Inference Time (s)
expresso	5	0.921 ± 0.001	0.399 ± 0.006	0.506 ± 0.004	0.390 ± 0.003	0.606 ± 0.020	54.344 ± 1.131
	10	0.919 ± 0.001	0.390 ± 0.008	0.499 ± 0.006	0.383 ± 0.008	0.602 ± 0.024	54.016 ± 1.081
	20	0.919 ± 0.001	0.387 ± 0.009	0.494 ± 0.009	0.381 ± 0.009	0.564 ± 0.042	52.712 ± 2.023
	Baseline Naive LLM (25-shot, 5 turns ea.)	0.903 ± 0.004	0.293 ± 0.016	0.426 ± 0.018	0.323 ± 0.016	0.576 ± 0.031	164.868 ± 5.179
styletalk	5	0.927 ± 0.028	0.636 ± 0.062	0.673 ± 0.066	0.543 ± 0.085	0.665 ± 0.041	22.949 ± 0.336
	10	0.935 ± 0.027	0.651 ± 0.067	0.688 ± 0.078	0.567 ± 0.106	0.650 ± 0.049	20.577 ± 4.518
	20	0.941 ± 0.032	0.665 ± 0.068	0.676 ± 0.092	0.562 ± 0.146	0.631 ± 0.048	18.857 ± 5.808
	Baseline Naive LLM (25-shot, 5 turns ea.)	0.888 ± 0.003	0.238 ± 0.014	0.310 ± 0.017	0.249 ± 0.014	0.176 ± 0.033	157.795 ± 4.875

Table 6: Model performance varying the *Prefix vector length* parameter, i.e. the output size of the transformer mapping stack and the prompt size in tokens that the LLM sees as input. The standard deviation across 5 trials is reported. The value 10 was chosen for use in the final model.

Test Set	Dialogue Length	BERTScore F1	METEOR	ChrF	ROUGE-L	Tag F1	Inference Time (s)
expresso	1	0.919 ± 0.001	0.388 ± 0.003	0.489 ± 0.005	0.381 ± 0.004	0.571 ± 0.016	39.067 ± 1.087
	3	0.919 ± 0.001	0.390 ± 0.008	0.499 ± 0.006	0.383 ± 0.008	0.602 ± 0.024	50.787 ± 2.418
	5	0.920 ± 0.001	0.392 ± 0.007	0.502 ± 0.006	0.384 ± 0.005	0.552 ± 0.013	64.932 ± 2.442
	<i>Baseline Naive LLM</i> (25-shot, 5 turns ea.)	0.891 ± 0.021	0.248 ± 0.081	0.378 ± 0.089	0.280 ± 0.076	0.471 ± 0.191	165.234 ± 70.581
styletalk	1	0.970 ± 0.017	0.744 ± 0.049	0.786 ± 0.051	0.696 ± 0.085	0.679 ± 0.010	14.156 ± 3.072
	3	0.935 ± 0.027	0.651 ± 0.067	0.688 ± 0.078	0.567 ± 0.106	0.650 ± 0.049	18.898 ± 4.164
	5	0.946 ± 0.029	0.655 ± 0.093	0.694 ± 0.082	0.590 ± 0.123	0.607 ± 0.052	17.970 ± 5.363
	<i>Baseline Naive LLM</i> (25-shot, 5 turns ea.)	0.885 ± 0.014	0.221 ± 0.051	0.297 ± 0.055	0.235 ± 0.047	0.164 ± 0.089	157.374 ± 64.981

Table 7: Model performance varying the *Dialogue Length* in the conversational context window used as input for the model. The standard deviation across 5 trials is reported. The value 3 was chosen for use in the final model.