Unlocking Fine-Grained and Within-Utterance Speaking Style Control in Prompt-Based Text-to-Speech Models

Anonymous ACL submission

Abstract

While prompt-based text-to-speech (TTS) models enable natural language-driven speaking style control, they often provide limited fine-grained control and apply a single global style across an utterance. This restricts practical use cases that require continuous style attribute interpolation across utterances and time-varying style transitions within a single utterance. In this paper, we propose novel techniques to achieve both capabilities in existing prompt-based TTS models. For inter-utterance style interpolation, we compute direction vectors between contrastive style prompts in the embedding space and perform simple interpolation, enabling smooth transitions between style characteristics. For intra-utterance style transition, we first identify a strong attention bias toward early tokens in autoregressive TTS decoders, causing the initial audio realization to dominate subsequent generation. To mitigate this effect, we introduce KV-cache swapping and sliding-window attention masking. Experiments demonstrate that our proposed inter-utterance interpolation achieves a 99-100% success rate in gender conversion, up to 36 Hz pitch variation, and up to 1.6 syllables-per-second speed change. Our intra-utterance transition maintains a speaker similarity of 0.81–0.91 and achieves perceptual smoothness scores of 3.48–4.48.

1 Introduction

Recent advances in text-to-speech (TTS) synthesis have introduced prompt-conditioned models that accept natural-language descriptions of speaking style (Guo et al., 2023; Yang et al., 2024; Lacombe et al., 2024). This approach offers a flexible interface for convenient and expressive voice generation; instead of preparing reference audio or selecting from a fixed set of style presets, users can describe intent directly in text (e.g., "calm male voice", "fast and high-pitched speech").

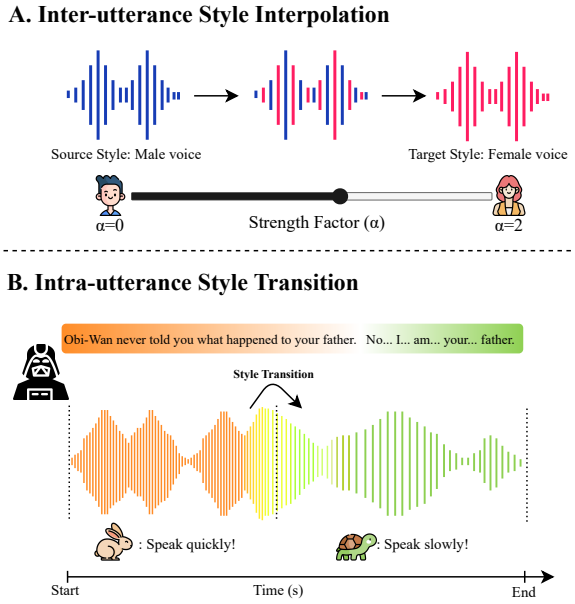


Figure 1: Overview of our training-free style control methods. (A) **Inter-utterance style interpolation** provides continuous control between source and target styles by adjusting the strength factor α. (B) **Intra-utterance style transition** unlocks style transitions within a single utterance during generation.

Despite this progress, achieving fine-grained and predictable control from natural language style prompts remains challenging. Many speech attributes, such as pitch and speed, vary continuously. However, current prompt-conditioned TTS models mostly accept coarse, discrete categorical terms (e.g., "fast", "slightly fast", "very fast", etc.) and small prompt edits do not reliably produce smooth, monotonic, and predictable changes in acoustic attributes (Korotkova et al., 2024; Ji et al., 2025). This is partially because current models assume a fixed global style condition (Leng et al., 2023; Yang et al., 2024; Ji et al., 2024), generating speech that is stylistically consistent but difficult to steer at a finer resolution.

These limitations are critical because many real-world applications require both continuous and

time-varying control. Audiobook narration benefits from gradual changes in speed or pitch to convey tension and emphasis (Guo et al., 2024). Conversational agents may need to shift tone within a single response to match discourse structure (Liu et al., 2024). Motivated by such use cases, we focus on two complementary controllability goals that are not well supported by prompt-based TTS models: (1) *inter-utterance style interpolation*: continuous control between contrastive style attributes (e.g., selecting an intermediate speaking rate between fast and slow) and (2) *intra-utterance style transition*: style transition within a single utterance (e.g., starting fast and gradually slowing down).

In this paper, we propose training-free methods that achieve both continuous inter-utterance controllability and intra-utterance style transition for natural-language-based TTS, as illustrated in Figure 1. We discover that existing models implicitly contain such controllability that can be *unlocked* via inference-time interventions. For inter-utterance control, we show that a simple representation-level approach is sufficient. Interpolating between style embeddings induced by contrastive prompts allows for finer control than discrete prompt edits.

In contrast, intra-utterance style transition reveals a qualitatively different obstacle. We identify a previously unreported phenomenon, termed **style self-referencing**: the first few seconds of generated speech disproportionately govern subsequent generation and can substantially reduce the effect of the natural-language style prompt. Consequently, naïve inference-time solutions, such as interpolating style embeddings or swapping prompts during generation, fail to produce reliable intra-utterance transitions.

Based on this analysis, we introduce novel inference-time mechanisms that directly counteract the self-referencing. Specifically, we employ (1) key-value (KV) cache swap and (2) sliding-window self-attention masking, both aiming to reduce the dominance of early-generated tokens. Together, these unlock smooth intra-utterance style transitions without fine-tuning.

Our contributions are as follows:

- We introduce two complementary forms of controllability in natural language-conditioned TTS: inter-utterance style control and intra-utterance style transition.
- We identify and characterize self-referencing in autoregressive TTS generation, where early-generated speech dominates later segments and overrides natural language guidance.
- Our proposed methods are entirely training-free, achieving fine-grained style control through inference-time techniques.
- We provide comprehensive evaluation demonstrating effective continuous control over pitch, speed, and gender attributes while maintaining speech quality.

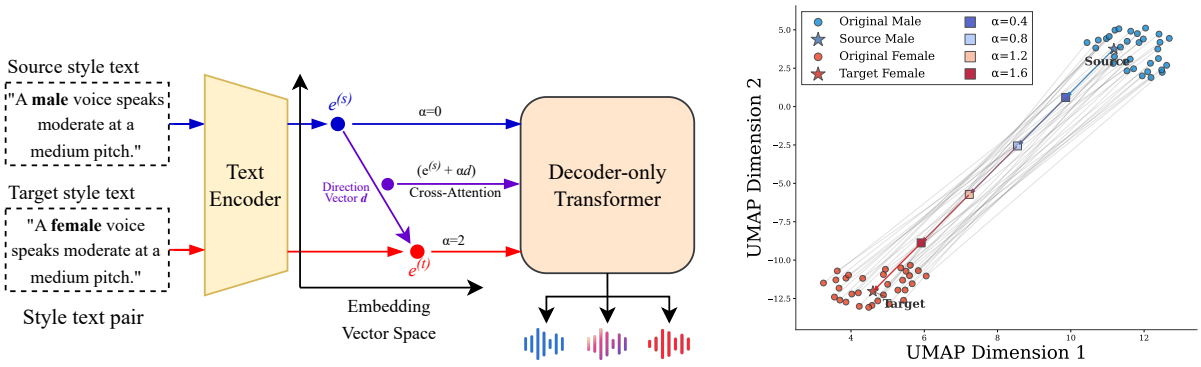
2 Related Work

2.1 Style Controllable Text-to-Speech

The pursuit of controllable TTS has primarily evolved through two paradigms: reference-based conditioning and, more recently, natural language-based prompting.

Reference-based approaches. Early TTS models conditioned synthesis on reference audio to capture style attributes. Global Style Tokens (GST) (Wang et al., 2018) and VAE-based methods (Zhang et al., 2019; Hsu et al., 2019) model prosodic variations as continuous latent variables, enabling style interpolation in the latent space. More recently, large-scale autoregressive models such as VALL-E (Wang et al., 2023; Chen et al., 2024), CosyVoice (Du et al., 2024, 2025), NaturalSpeech 3 (Ju et al., 2024), and Spark-TTS (Wang et al., 2025b) achieve human-parity synthesis by conditioning on reference speech. Despite their fidelity, these approaches require reference audio, which prevents users from synthesizing specific styles in the absence of matching acoustic samples.

Natural language-based approaches. To overcome the reliance on reference audio, recent research has shifted towards controlling TTS via natural language descriptions. PromptTTS (Guo et al., 2023) pioneers this by learning a mapping between textual descriptions and acoustic style latents, enabling attribute control (e.g., gender, pitch, speed) through prompts. This line of work has been rapidly extended by models such as InstructTTS (Yang et al., 2024), which employs cross-modal metric learning to follow free-form instructions, and Parler-TTS (Lyth and King, 2024), which leverages large-scale synthetic annotations to achieve high-fidelity description-guided synthesis. However, these models generally treat style prompts as static characteristics applied globally to the entire



(a) Overview of the inter-utterance style interpolation.

(b) U-MAP visualization.

Figure 2: **Inter-utterance Style Interpolation.** (a) Given a source-target style prompt pair, we extract their embeddings $\mathbf{e}^{(s)}$ and $\mathbf{e}^{(t)}$ from the text encoder and compute the direction vector $\mathbf{d} = \mathbf{e}^{(t)} - \mathbf{e}^{(s)}$. By varying the interpolation strength α, we generate new style embeddings to continuously control the speaking style attributes. (b) UMAP visualization of the embedding space shows that interpolated embeddings (colored points with $\alpha \in \{0.4, 0.8, 1.2, 1.6\}$) form a smooth trajectory between source (male) and target (female) style representations.

utterance. Current architectures often lack mechanisms to interpret time-varying prompts (e.g., "start calmly and become excited") or to continuously modulate style intensity within an utterance, resulting in largely monotonic expressivity.

2.2 Fine-Grained Speaking Style Control

Significant efforts have progressed from predicting prosodic features via variance adaptors in non-autoregressive models like FastSpeech-2 (Ren et al., 2021) and FastPitch (Łańcucki, 2021) to more explicit manipulation of intra-utterance dynamics. Recent advancements such as Lina-Style (Lemerle et al., 2025) and WeSCon (Wang et al., 2025a) achieve word-level emotional control through synthetic data interleaving and multi-stage inference, respectively. Furthermore, models like ELaTE (Kanda et al., 2024) and EmoCtrl-TTS (Wu et al., 2024) extend this control to non-verbal vocalizations and continuous arousal-valence trajectories, allowing for a more dynamic emotional flow than traditional text-driven predictions.

The integration of LLMs allows intuitive, zero-shot emotion control through natural-language prompts, as explored in PUE (Gao et al., 2025). However, while recent training-free methods like EmoSteer-TTS (Xie et al., 2025) enable fine-grained emotion modulation via activation steering, they focus primarily on manipulating distinct emotional attributes rather than ensuring smooth, continuous style transitions across time. Consequently, achieving natural *intra-utterance* style transitions remains an open challenge.

3 Inter-Utterance Style Interpolation

In this section, we present our method to achieve continuous inter-utterance style control by manipulating direction vectors between contrastive style prompts in the style embedding space.

3.1 Analysis: Style Vectors in Embedding Space

To investigate the characteristics of natural language style embeddings, we analyze how contrastive style attributes are represented in the embedding space. Specifically, we prepare multiple pairs of style prompts that differ only in a single target attribute. For example, for gender control, we use pairs like "A **male** voice with clean audio" and "A **female** voice with clean audio", where only the gender token varies while other attributes in the prompt remain identical. We encode these prompts using the prompt text encoder and extract only the embeddings corresponding to the target style attribute tokens (e.g., "male" and "female"). To visualize their distribution, we project these attribute embeddings into a lower-dimensional space using UMAP (McInnes et al., 2018).

As shown in Figure 2b, we observe that embeddings of the same attribute form tight clusters: all "male" tokens cluster together, and all "female" tokens form a separate cluster. Importantly, these contrastive attribute clusters (e.g., "male" vs. "female") are well-separated in the embedding space. This clear separation suggests that linear interpolation between contrastive attribute vectors can produce continuous style transitions.

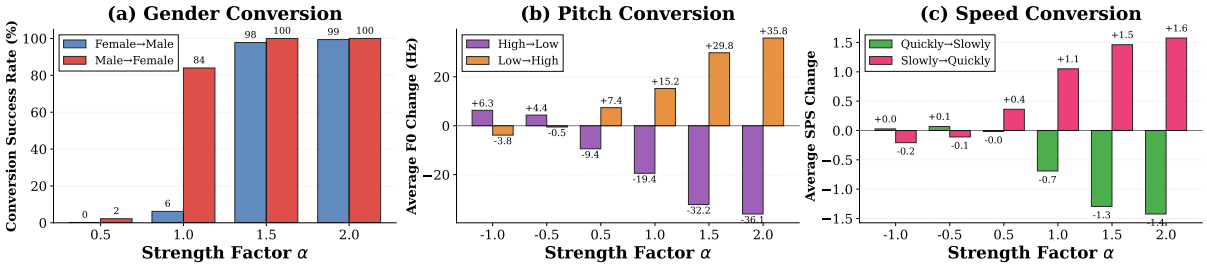


Figure 3: Inter-utterance style interpolation results. (a) Gender conversion success rate, (b) average F0 change for pitch control, (c) average speaking rate change for speed control. All attributes show monotonic changes with α in the positive range, demonstrating continuous controllability.

Attribute	Direction	Success	Δ Metric	SIM
Gender	Female → Male	99.0%	–	–
	Male → Female	100.0%	–	–
Pitch	High → Low	96.3%	−36.1 Hz	0.76
	Low → High	93.0%	+35.8 Hz	0.78
Speed	Quick → Slow	94.2%	−1.4 SPS	0.84
	Slow → Quick	95.7%	+1.6 SPS	0.84

Table 1: Objective evaluation of inter-utterance style interpolation.

Attr.	Direction	Style Change (α)			MOS
		0.5	1.0	2.0	
Gender	Female → Male	0.09	0.12	1.74	4.25
	Male → Female	0.18	0.62	2.00	4.40
Pitch	High → Low	0.95	1.09	1.71	4.26
	Low → High	0.68	1.18	1.62	4.29
Speed	Quick → Slow	0.26	0.75	1.62	3.99
	Slow → Quick	0.43	1.45	1.77	4.32

Table 2: Subjective evaluation of inter-utterance style interpolation.

3.2 Method

We implement a simple embedding vector interpolation method for continuous style control. Let $\boldsymbol{E}^{(s)} = \{\boldsymbol{e}_1^{(s)}, \boldsymbol{e}_2^{(s)}, ..., \boldsymbol{e}_l^{(s)}\}$ and $\boldsymbol{E}^{(t)} = \{\boldsymbol{e}_1^{(t)}, \boldsymbol{e}_2^{(t)}, ..., \boldsymbol{e}_l^{(t)}\}$ denote the text encoder outputs for the source and target style prompts, respectively. Since the two prompts differ only in the attribute tokens, we compute the direction vector for the set of attribute token positions, $\mathcal{A}$. Then, the direction vector is computed as below:

$$\boldsymbol{d}_i = \frac{1}{2}\left(\boldsymbol{e}_i^{(t)} - \boldsymbol{e}_i^{(s)}\right), \quad i \in \mathcal{A} \tag{1}$$

We apply the direction vector only to the positions corresponding to the attribute tokens. The interpolated embedding is computed as:

$$\boldsymbol{e}'_i = \begin{cases} \boldsymbol{e}_i^{(s)} + \alpha \cdot \boldsymbol{d}_i & \text{if } i \in \mathcal{A} \\ \boldsymbol{e}_i^{(s)} & \text{otherwise} \end{cases} \tag{2}$$

where $\alpha \in \mathbb{R}$ is the interpolation strength. When $\alpha = 0$, the output reproduces the source style; when $\alpha = 2$, it produces the target style; and intermediate values yield smoothly interpolated styles. Values outside $[0, 2]$ correspond to extrapolation beyond the original style range.

Figure 2a illustrates our method. Given a source-target style text pair, we extract their embeddings from the text encoder and compute the direction vector between the contrastive tokens. By adding the scaled direction vector $\alpha \cdot \boldsymbol{d}$ to the source embedding, we obtain the interpolated style representation $\boldsymbol{E}'$. This representation is then fed to the decoder through cross-attention to generate speech with the desired style characteristics. We empirically verify that applying interpolation to all tokens yields similar performance to our attribute-only approach (see Appendix B).

3.3 Experimental Results

Setup. We evaluate our inter-utterance style interpolation on three attributes: gender, pitch, and speed. We take 400 sentences from the LibriTTS-R (Koizumi et al., 2023) test set and generate speech using the Parler-TTS-mini model (Lacombe et al., 2024; Lyth and King, 2024). Further details are provided in the Appendix A.

Objective Evaluation. Figure 3 presents that our method achieves effective style control across all tested attributes. For gender conversion, the success rate increases with α, achieving near-complete conversion at $\alpha \geq 1.5$ (98% for Female→Male, 100% for Male→Female). For pitch and speed, which are continuous attributes, our method enables smooth and gradual control. Pitch (F0) changes

linearly with α, achieving up to 36.1 Hz decrease (High→Low) and 35.8 Hz increase (Low→High) at $\alpha = 2.0$, with intermediate values producing perceptibly distinct pitch levels. Similarly, speaking rate varies proportionally with α; Quick→Slow reduces syllable-per-second (SPS) (Wang et al., 2025b) by 0.7 at $\alpha = 1.0$ and 1.4 at $\alpha = 2.0$, while Slow→Quick increases SPS by 1.1 and 1.6, respectively, demonstrating fine-grained continuous control. Table 1 summarizes the results at $\alpha = 2.0$.

Subjective Evaluation. In Table 2, we show human evaluation results where participants rated both style change (-2 to $+2$) and naturalness (MOS, 1–5) (see Appendix A.3 for details). For continuous attributes (pitch and speed), participants consistently rated intermediate interpolation strengths ($\alpha = 0.5, 1.0$) lower than full conversion ($\alpha = 2.0$), confirming gradual style transitions. At $\alpha = 2.0$, all conversion directions achieved strong perceived style change (scores 1.62–2.00) while maintaining high naturalness (MOS above 3.99).

4 Intra-utterance Style Transition

In this section, we address the novel challenge of varying style within a single utterance during autoregressive generation. We first describe our initial attempts and analyze why they fail, then present our solution.

4.1 Intuition: Why Naïve Approaches Fail

Building upon our inter-utterance style interpolation (Section 3), we initially attempted to achieve intra-utterance style variation during autoregressive generation in a similar manner. Specifically, we expected that switching the style embedding from $\boldsymbol{E}^{(s)}$ to $\boldsymbol{E}'$ at a transition point t^* would cause the autoregressive decoder to generate all subsequent tokens ($t > t^*$) with the target style.

However, this naïve approach fails to produce desired style transitions. In fact, the generated speech *continues to exhibit the initial style characteristics* even after modifying the style embeddings. This unexpected behavior, where the same manipulation that works across utterances fails within a single utterance, motivated us to investigate the decoder's attention patterns.

Early-Token Bias in Cross-Attention. To understand why modifying style representations mid-generation fails, we analyze the cross-attention patterns between generated audio tokens and style text tokens. Figure 4 visualizes the cross-attention weights in the autoregressive decoder across multiple layers, where rows correspond to style text tokens and columns correspond to audio token positions during generation.

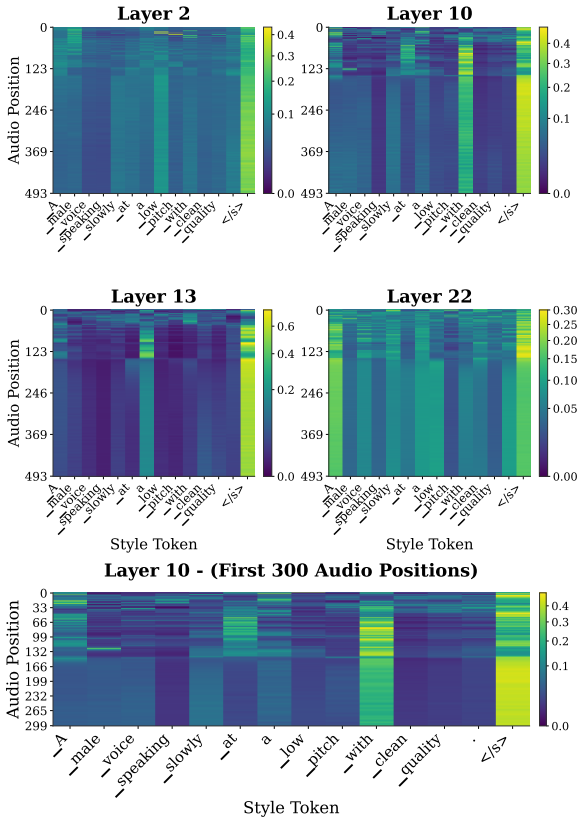


Figure 4: Cross-attention weights between style text tokens (rows) and generated audio tokens (columns) across multiple decoder layers. Attention weights are actively updated during early generation, then remain nearly constant throughout subsequent generation.

In the early generation phase, the decoder actively attends to style tokens with dynamically changing attention patterns. This indicates that the model incorporates style information from the text prompt to establish the acoustic characteristics of the speech. In contrast, in the following generation phase, the attention weights across all style tokens become fixed and show minimal variation. Rather than continuing to query the style representation, the decoder maintains a consistent attention distribution focusing on relatively less informative (e.g., "with", "<EOS>") tokens.

This observation suggests that prompt-based autoregressive TTS models follow a "set-and-maintain" strategy: the model establishes style characteristics during initial generation and subsequently maintains consistency by relying on these

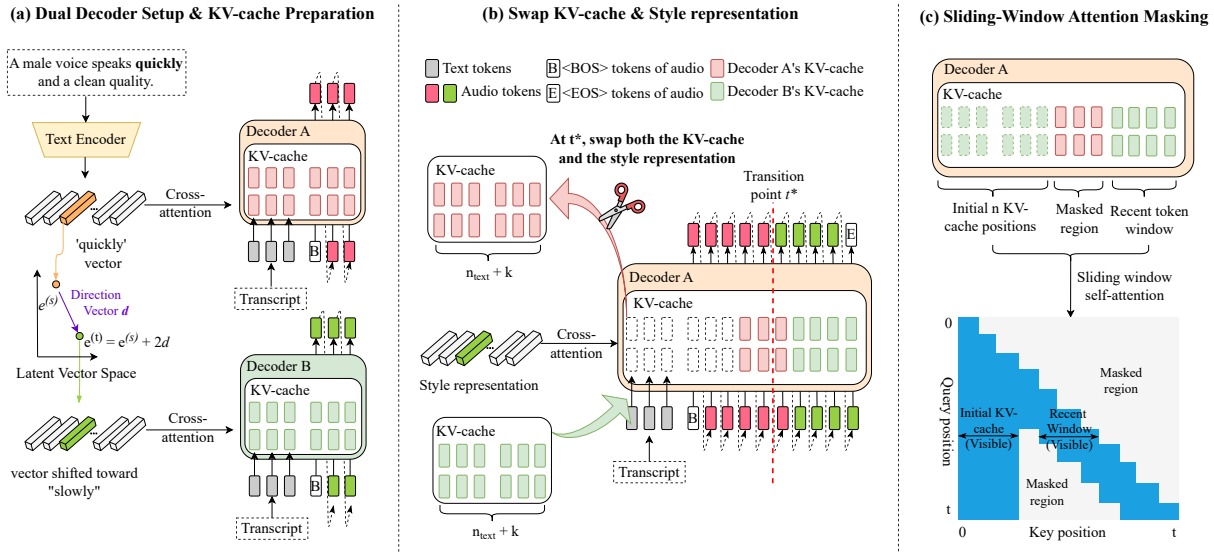


Figure 5: **Intra-utterance style transition.** To convert speaking style mid-generation, we combine multiple techniques to avoid style self-referencing: (a) prepare an initial KV-cache derived from the target style prompt, and (b) swap the original KV-cache and style embeddings, together with (c) sliding-window attention masking.

early acoustic representations through autoregressive (i.e., self-attention) conditioning. We term this phenomenon **style self-referencing**. Once the style is encoded in the early audio tokens, the decoder becomes resistant to new style information provided through cross-attention.

This early-token attention bias explains why our initial approach fails: even when we replace the style representation at t^*. Therefore, achieving intra-utterance style variation requires not only modifying the cross-attention style input but also *addressing the persistent influence of early-generated tokens* on autoregressive generation.

4.2 Method

Motivated by this analysis, we combine KV-cache swapping and sliding-window attention masking to enable intra-utterance style transitions. Figure 5 illustrates the overall process.

Dual Decoder Setup. To enable style transitions during generation, we prepare two decoder instances. *Decoder-A* generates speech in source style, conditioned on the original style representation $\boldsymbol{E}^{(s)}$ until the desired transition point t^*. In parallel, *Decoder-B* generates the initial n (where $n \ll t^*$) tokens to construct KV-cache for the target style, conditioned on the modified style representation $\boldsymbol{E}'$ obtained through vector interpolation (Section 3). Note that the additional cost of Decoder-B is marginal, as most of the latency comes from autoregressive token generation.

KV-Cache Swap. As the generation reaches the desired transition point t^*, we swap the initial KV-cache in Decoder-A with that from Decoder-B for *initial* n *positions* across all layers:

$$\boldsymbol{K}^{(A)}_{1:n}, \boldsymbol{V}^{(A)}_{1:n} \leftarrow \boldsymbol{K}^{(B)}_{1:n}, \boldsymbol{V}^{(B)}_{1:n}. \tag{3}$$

We set $n = n_{\text{text}} + k$. Here, n_{text} is the number of tokens in the text prompt (the linguistic content to be spoken), and k is an additional buffer that covers the early audio tokens where style characteristics are actively encoded during initial generation. This initial region is critical because, as discussed in our analysis, the decoder establishes style information not only through the prompt representation but also through these early generated tokens.

Simultaneously, we also replace the style representation used in cross-attention with the modified version $\boldsymbol{E}'$. This ensures that both the initial KV-cache and the incoming style information reflect the target style.

Sliding-window Attention Masking. In addition, we discover that KV-cache swap alone is insufficient because standard self-attention allows the decoder to attend to all previously generated tokens. This means that even after replacing the initial region with target style KV-cache, the decoder can still access the original style information encoded in tokens from positions in-between ($n + 1$ to t^*).

Attribute	Direction	Window Size	Δ Metric	SIM	Trans. (%)	Smoothness
Pitch	High → Low	256	**−12.4 Hz**	0.81	**96.2**	4.20
		384	−8.97 Hz	0.85	73.1	3.79
		512	−10.9 Hz	0.87	80.0	4.20
		Full	−11.5 Hz	**0.90**	55.6	4.00
	Low → High	256	**+27.4 Hz**	0.84	**96.2**	3.48
		384	+19.6 Hz	0.87	73.1	3.79
		512	+16.6 Hz	0.88	80.0	4.20
		Full	+5.5 Hz	**0.91**	57.7	4.00
Speed	Quick → Slow	256	**−2.29 SPS**	0.81	**88.0**	3.82
		384	−1.74 SPS	0.84	80.8	3.82
		512	−1.93 SPS	0.86	80.8	4.48
		Full	−1.34 SPS	**0.90**	77.8	4.33
	Slow → Quick	256	**+1.02 SPS**	0.86	**92.0**	3.87
		384	+0.74 SPS	0.87	**92.0**	4.00
		512	+0.69 SPS	0.89	76.0	3.79
		Full	+0.54 SPS	**0.91**	84.6	4.18

Table 3: Intra-utterance style transition results. Δ Metric indicates the average pitch (Hz) or speed (SPS) difference between the first and last 3-second segments. SIM indicates speaker similarity between the two segments. Trans. indicates the percentage of samples where listeners perceived a style transition. Smoothness rates the naturalness of style transition on a 1–5 scale. Results are reported using a KV-cache buffer size of $k = 48$.

These intermediate tokens, already generated under the source style, would continue to influence subsequent generation and prevent effective style transition.

To address this issue, we introduce sliding-window attention masking (Beltagy et al., 2020; Xiao et al., 2024) that restricts self-attention to only two specific regions: (1) the initial n positions containing target style KV-cache and (2) the most recent w tokens in the local window.

Formally, let i denote the current query position, j denote the key position, n denote the size of the replaced initial KV region, and w denote the sliding-window size. For a token at position $i > t^*$ (after the style transition point), the attention mask is defined as:

$$M_{ij} = \begin{cases} 0 & \text{if } j \le n,\ i - w \le j \le i \\ -\infty & \text{otherwise} \end{cases} \quad (4)$$

where $M_{ij} = 0$ allows attention and $M_{ij} = -\infty$ blocks attention after softmax.

By attending only to the replaced initial region and recent tokens, the decoder gradually adopts the new style characteristics while maintaining local coherence. Note that local coherence is crucial for the overall quality and naturalness of the generated speech (Ye et al., 2025). Tokens generated immediately after t^* are influenced by both the new style (from the replaced initial region) and the local context (from the recent window), enabling smooth style progression rather than abrupt transitions. See the Appendix E for the full algorithm.

4.3 Experimental Results

Setup. To generate sufficiently long speech for observing style transitions, we select 400 samples from the LibriTTS-R test set with text token lengths between 50 and 70. We compare different sliding-window sizes $w \in \{256, 384, 512, \text{Full}\}$ to analyze the effect of attention scope on style transition effectiveness and speech quality, where "Full" denotes standard self-attention without sliding-window masking. See Section 4.4 for the ablation study on the KV-cache size k.

Objective Results. Table 3 presents the results for different window sizes and style attribute changes. As expected, smaller windows produce more pronounced changes: window size of 256 achieves $+27.4$ Hz for Low→High pitch conversion (vs. $+5.5$ Hz for Full) and -2.29 SPS for Quick→Slow speed conversion (vs. -1.34 SPS for Full). However, speaker similarity decreases with smaller windows (0.81 for size 256 vs. 0.90–0.91 for Full), demonstrating a trade-off between transition strength and identity preservation.

Subjective Results. Subjective evaluations confirm this trade-off. For the perceptual transition

Attribute	Direction	Style diff	SIM
Pitch	High → Low	−4.3 Hz	0.92
	Low → High	−2.40 Hz	0.92
Speed	Quick → Slow	−0.23 SPS	0.93
	Slow → Quick	−0.27 SPS	0.91

Table 4: Style transition results when only replacing the style embedding without KV-cache swap.

detection, window size of 256 achieves the highest rate (88.0–96.2% across attributes), while Full attention showed lower detection rates (55.6–84.6%). For smoothness, all configurations achieved scores above 3.48, with larger windows (512, Full) reaching up to 4.48, implying that more gradual transitions appear more natural.

Effect of Window Size. This trade-off arises from the attention mechanism's access to style information. After the KV-cache swap at t^*, intermediate tokens (positions $n+1$ to t^*) retain the original source style. Smaller windows rapidly forget these intermediate source-style tokens, minimizing their influence on subsequent generation and enabling stronger, more abrupt style transitions. In contrast, larger windows allow the decoder to attend to more intermediate tokens for longer, producing gradual blending that appears smoother but weakens the transition effect.

4.4 Analysis

Style Replacement Without KV-Cache Swap. In Table 4, we investigate whether simply replacing the style representation mid-generation is sufficient for style transition, without any KV-cache swap. The results show that replacing only the style representation produces minimal style change. This confirms our analysis in Section 4.1; due to the early-token bias, the model continues to attend to the cached style information from the initial tokens, making the newly introduced style representation ineffective. This demonstrates that the proposed KV-cache swap is essential for achieving intra-utterance style transitions.

Joint Effect of Window and KV-Cache Size. Our method has two key hyperparameters: the sliding-window size w and the KV-cache region size $n = n_{\text{text}} + k$. We conduct a grid search over window sizes ($w \in \{256, 384, 512\}$) and additional KV-cache tokens ($k \in \{0, 32, 48\}$) to analyze their joint effect on style transition.

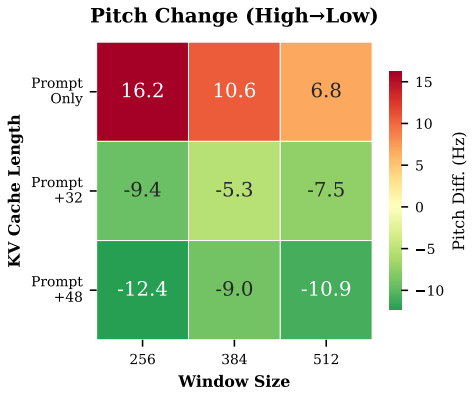


Figure 6: Ablation study on window size and KV-cache region size ($n = n_{\text{text}} + k$) for High→Low pitch conversion.

Figure 6 visualizes the pitch difference (High→Low) for each configuration. Negative values indicate successful pitch reduction toward the target style. The results validate two key design decisions in our method. When swapping only the text region ($k = 0$), all window sizes produce reversed pitch changes (+6.8 to +16.2 Hz), indicating that the KV-cache corresponding to the text region alone lacks sufficient target style information. However, extending the swap to include the initial KV-cache ($k \geq 32$) for acoustic tokens enables successful conversion across all window sizes (−5.3 to −12.4 Hz). This supports our findings that the model encodes critical style information in the initial KV-cache positions beyond the prompt, and our KV-cache swap must include these positions (i.e., $k > 0$) to achieve effective transitions.

5 Conclusion

We presented a training-free approach for fine-grained speaking-style control in prompt-based TTS, addressing both inter-utterance style interpolation and intra-utterance style transition. For inter-utterance control, we showed that continuous attribute intensity can be adjusted smoothly by interpolating along style directions in the text encoder's embedding space, improving monotonicity and allowing precise control without degrading speech quality. For intra-utterance control, we identified a style self-referencing effect in autoregressive decoding that limits mid-utterance prompt changes. We proposed a method to mitigate this issue using KV-cache swapping and sliding-window attention masking to produce clearer, smoother transitions while largely preserving the speaker identity.

Limitations

This work focuses on three attributes (pitch, speed, and gender), but there is room for extension to other style attributes such as emotion and intonation. Additionally, there is a trade-off depending on the window size in the intra-utterance style transition. Smaller windows produce more pronounced style transitions but reduce speaker similarity, while larger windows better preserve speech quality but weaken the desired effect; further improvements to mitigate this trade-off remain a direction for future work. While the additional memory and computational costs are minimal, intra-utterance style transition requires two decoder inferences; practicality could be improved through lightweight approaches. Meanwhile, the proposed methods have only been validated on natural language prompt-based autoregressive TTS models, and their applicability to non-autoregressive or diffusion-based models that lack KV-cache structures remains an open question. Finally, although training-free continuous style control for prompt-based TTS is the objective of this research, there are opportunities for extending this to training-based approaches.

Ethics Statements

While our proposed style control methods do not aim to mimic the real person's voice and focus on enhancing the controllability of speech synthesis, the domain requires ethical consideration regarding potential misuse. Continuous style interpolation techniques could be exploited to imitate or manipulate a specific speaker's voice, potentially leading to fraud, disinformation, or non-consensual voice cloning. To prevent such risks, detection technologies that distinguish synthetic speech from real speech, as well as mechanisms such as audio watermarking, should be developed in parallel.

References

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Sanyuan Chen, Shujie Liu, Long Zhou, Yanqing Liu, Xu Tan, Jinyu Li, Sheng Zhao, Yao Qian, and Furu Wei. 2024. Vall-e 2: Neural codec language models are human parity zero-shot text to speech synthesizers. *arXiv preprint arXiv:2406.05370*.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, and 1 others. 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.

Zhihao Du, Qian Chen, Shiliang Zhang, Kai Hu, Heng Lu, Yexin Yang, Hangrui Hu, Siqi Zheng, Yue Gu, Ziyang Ma, and 1 others. 2024. Cosyvoice: A scalable multilingual zero-shot text-to-speech synthesizer based on supervised semantic tokens. *CoRR*.

Zhihao Du, Changfeng Gao, Yuxuan Wang, Fan Yu, Tianyu Zhao, Hao Wang, Xiang Lv, Hui Wang, Chongjia Ni, Xian Shi, and 1 others. 2025. Cosyvoice 3: Towards in-the-wild speech generation via scaling-up and post-training. *arXiv preprint arXiv:2505.17589*.

Xiaoxue Gao, Huayun Zhang, and Nancy F Chen. 2025. Prompt-unseen-emotion: Zero-shot expressive speech synthesis with prompt-llm contextual knowledge for mixed emotions. *arXiv preprint arXiv:2506.02742*.

Dake Guo, Xinfa Zhu, Liumeng Xue, Yongmao Zhang, Wenjie Tian, and Lei Xie. 2024. Text-aware and Context-aware Expressive Audiobook Speech Synthesis. In *Interspeech 2024*, pages 1790–1794.

Zhifang Guo, Yichong Leng, Yihan Wu, Sheng Zhao, and Xuejiao Tan. 2023. Prompttts: Controllable text-to-speech with text descriptions. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Wei-Ning Hsu, Yu Zhang, Ron Weiss, Heiga Zen, Yonghui Wu, Yuan Cao, and Yuxuan Wang. 2019. Hierarchical generative modeling for controllable speech synthesis. In *International Conference on Learning Representations*.

Shengpeng Ji, Qian Chen, Wen Wang, Jialong Zuo, Minghui Fang, Ziyue Jiang, Hai Huang, Zehan Wang, Xize Cheng, Siqi Zheng, and 1 others. 2025. Controlspeech: Towards simultaneous and independent zero-shot speaker cloning and zero-shot language style control. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6966–6981.

Shengpeng Ji, Jialong Zuo, Minghui Fang, Ziyue Jiang, Feiyang Chen, Xinyu Duan, Baoxing Huai, and Zhou Zhao. 2024. Textrolspeech: A text style control speech corpus with codec language text-to-speech models. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 10301–10305. IEEE.

Zeqian Ju, Yuancheng Wang, Kai Shen, Xu Tan, Detai Xin, Dongchao Yang, Eric Liu, Yichong Leng, Kaitao Song, Siliang Tang, and 1 others. 2024. Naturalspeech 3: Zero-shot speech synthesis with factorized codec and diffusion models. In *International Conference on Machine Learning*, pages 22605–22623. PMLR.

Naoyuki Kanda, Xiaofei Wang, Sefik Emre Eskimez, Manthan Thakker, Hemin Yang, Zirun Zhu, Min Tang, Canrun Li, Chung-Hsien Tsai, Zhen Xiao, and 1 others. 2024. Making flow-matching-based zero-shot text-to-speech laugh as you like. *arXiv preprint arXiv:2402.07383*.

Yuma Koizumi, Heiga Zen, Shigeki Karita, Yifan Ding, Kohei Yatabe, Nobuyuki Morioka, Michiel Bacchiani, Yu Zhang, Wei Han, and Ankur Bapna. 2023. Libritts-r: A restored multi-speaker text-to-speech corpus. *Interspeech 2023*.

Yuliya Korotkova, Ilya Kalinovskiy, and Tatiana Vakhrusheva. 2024. Word-level text markup for prosody control in speech synthesis. In *Proc. Interspeech 2024*, pages 2280–2284.

Yoach Lacombe, Vaibhav Srivastav, and Sanchit Gandhi. 2024. Parler-tts. https://github.com/huggingface/parler-tts.

Adrian Łańcucki. 2021. Fastpitch: Parallel text-to-speech with pitch prediction. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6588–6592. IEEE.

Théodor Lemerle, Nicolas Obin, and Axel Roebel. 2025. Lina-style: Word-level style control in tts via interleaved synthetic data. In *Proc. SSW 2025*, pages 35–39.

Yichong Leng, Zhifang Guo, Kai Shen, and 1 others. 2023. Prompttts 2: Describing and generating voices with text prompt. *arXiv preprint arXiv:2309.02285*.

Rui Liu, Yifan Hu, Yi Ren, Xiang Yin, and Haizhou Li. 2024. Emotion rendering for conversational speech synthesis with heterogeneous graph-based context modeling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18698–18706.

Dan Lyth and Simon King. 2024. Natural language guidance of high-fidelity text-to-speech with synthetic annotations. *arXiv preprint arXiv:2402.01912*.

Leland McInnes, John Healy, and James Melville. 2018. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*.

Max Morrison, Caedon Hsieh, Nathan Pruyne, and Bryan Pardo. 2023. Cross-domain neural pitch and periodicity estimation. *arXiv preprint arXiv:2301.12258*.

Yi Ren, Chenxu Hu, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. 2021. Fastspeech 2: Fast and high-quality end-to-end text to speech. In *International Conference on Learning Representations*.

Chengyi Wang, Sanyuan Chen, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, and 1 others. 2023. Neural codec language models are zero-shot text to speech synthesizers. *arXiv preprint arXiv:2301.02111*.

Tianrui Wang, Haoyu Wang, Meng Ge, Cheng Gong, Chunyu Qiang, Ziyang Ma, Zikang Huang, Guanrou Yang, Xiaobao Wang, EngSiong Chng, Xie Chen, Longbiao Wang, and Jianwu Dang. 2025a. Word-level emotional expression control in zero-shot text-to-speech synthesis. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Xinsheng Wang, Mingqi Jiang, Ziyang Ma, Ziyu Zhang, Songxiang Liu, Linqin Li, Zheng Liang, Qixi Zheng, Rui Wang, Xiaoqin Feng, and 1 others. 2025b. Spark-tts: An efficient llm-based text-to-speech model with single-stream decoupled speech tokens. *arXiv preprint arXiv:2503.01710*.

Yuxuan Wang, Daisy Stanton, Yu Zhang, RJ-Skerry Ryan, Eric Battenberg, Joel Shor, Ying Xiao, Ye Jia, Fei Ren, and Rif A Saurous. 2018. Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis. In *International conference on machine learning*, pages 5180–5189. PMLR.

Haibin Wu, Xiaofei Wang, Sefik Emre Eskimez, Manthan Thakker, Daniel Tompkins, Chung-Hsien Tsai, Canrun Li, Zhen Xiao, Sheng Zhao, Jinyu Li, and 1 others. 2024. Laugh now cry later: Controlling time-varying emotional states of flow-matching-based zero-shot text-to-speech. In *2024 IEEE Spoken Language Technology Workshop (SLT)*, pages 690–697. IEEE.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024. Efficient streaming language models with attention sinks. In *The Twelfth International Conference on Learning Representations*.

Tianxin Xie, Shan Yang, Chenxing Li, Dong Yu, and Li Liu. 2025. Emosteer-tts: Fine-grained and training-free emotion-controllable text-to-speech via activation steering. *arXiv preprint arXiv:2508.03543*.

Dongchao Yang and 1 others. 2024. Instructtts: Modelling expressive tts in discrete latent space with natural language style prompt. *arXiv preprint arXiv:2301.13662*.

Zhen Ye, Xinfa Zhu, Chi-Min Chan, Xinsheng Wang, Xu Tan, Jiahe Lei, Yi Peng, Haohe Liu, Yizhu Jin, Zheqi Dai, and 1 others. 2025. Llasa: Scaling train-time and inference-time compute for llama-based speech synthesis. *arXiv preprint arXiv:2502.04128*.

Ya-Jie Zhang, Shifeng Pan, Lei He, and Zhen-Hua Ling. 2019. Learning latent representations for style control and transfer in end-to-end speech synthesis. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6945–6949. IEEE.

A Experimental Details

A.1 Objective Evaluation Metrics

We use the following metrics for objective evaluation across both inter-utterance and intra-utterance experiments:

Pitch Measurement. We measure the average fundamental frequency (F0) in Hz using PENN (Pitch Estimator Neural Network) (Morrison et al., 2023).

Speed Measurement. We compute speaking rate as syllables per second (SPS): the total number of syllables divided by the utterance duration (Wang et al., 2025b).

Speaker Similarity. We extract speaker embeddings using a fine-tuned WavLM[1] (Chen et al., 2022) and compute cosine similarity between audio segments to verify that speaker identity is preserved.

Gender Classification. For gender conversion evaluation, we use a pre-trained gender classification model[2] to measure the conversion success rate.

A.2 Dataset and Model

For inter-utterance style interpolation, we selected 400 sentences from the LibriTTS-R (Koizumi et al., 2023) test set. For intra-utterance style variation, we selected 400 samples from the LibriTTS-R test set where the text token length is between 50 and 70 to ensure sufficiently long speech for observing style transitions. All audio samples were generated using the Parler-TTS-mini model (Lacombe et al., 2024; Lyth and King, 2024).

A.3 Inter-Utterance Style Interpolation

Objective Evaluation. We generate speech at various interpolation strengths and measure the change compared to the original style ($\alpha = 0$). For gender conversion, we evaluate $\alpha \in \{0.5, 1.0, 1.5, 2.0\}$, while for pitch and speed conversion, we extend the range to $\alpha \in \{-1.0, -0.5, 0.5, 1.0, 1.5, 2.0\}$ to examine both interpolation and extrapolation. We use the metrics described in Appendix A.1.

[1] microsoft/wavlm-base-plus-sv

[2] alefiury/wav2vec2-large-xlsr-53-gender-recognition-librispeech

Subjective Evaluation. We recruited 15 participants to evaluate the converted speech across different interpolation strengths $\alpha \in \{0.5, 1.0, 2.0\}$. Figure 8 shows the evaluation interface. For each sample, participants were presented with two audio clips: the source audio and the converted audio. The evaluation target (Gender, Pitch, or Speed) and conversion direction were clearly indicated. Participants could listen to both audio samples multiple times before rating two criteria. First, they evaluated the **Style Conversion Score**, indicating whether the converted audio changed toward the target style on a 5-point scale: -2 (opposite direction), -1 (slightly opposite), 0 (no change), $+1$ (slightly changed), and $+2$ (fully changed). Second, they rated the **Audio Quality (MOS)** on a 5-point scale ranging from 1 (Bad) to 5 (Excellent).

A.4 Intra-Utterance Style Transition

Objective Evaluation. To measure style transition effectiveness, we extract the first 3 seconds and the last 3 seconds of each generated utterance and compute the difference in style attributes between these segments using the metrics in Appendix A.1. Specifically, we calculate the **Pitch diff** (difference in average F0) and **Speed diff** (difference in speaking rate) between the two segments. Additionally, we compute **Speaker Similarity (SIM)** using cosine similarity between speaker embeddings to verify that speaker identity is preserved across the transition.

Subjective Evaluation. We evaluated both the detectability and smoothness of style transitions using the interface shown in Figure 9. For each sample, participants listened to a single audio clip containing the style transition, with the intended target (e.g., Speed) and direction (e.g., Quick $\rightarrow$ Slow) displayed. Participants answered two questions: **Q1 (Style Transition Detection)**, a binary choice of whether they perceived the speech changing in the intended direction; and **Q2 (Smoothness)**, rating the naturalness of the transition on a 5-point scale from 1 (Very Unnatural) to 5 (Very Natural). The detection rate is reported as Trans. (%) in Table 3.

B Full Vector vs. Attribute-Only Vector Interpolation

In Section 3, we apply the direction vector only to attribute token positions $i \in \mathcal{A}$, leaving other token embeddings unchanged. An alternative approach is

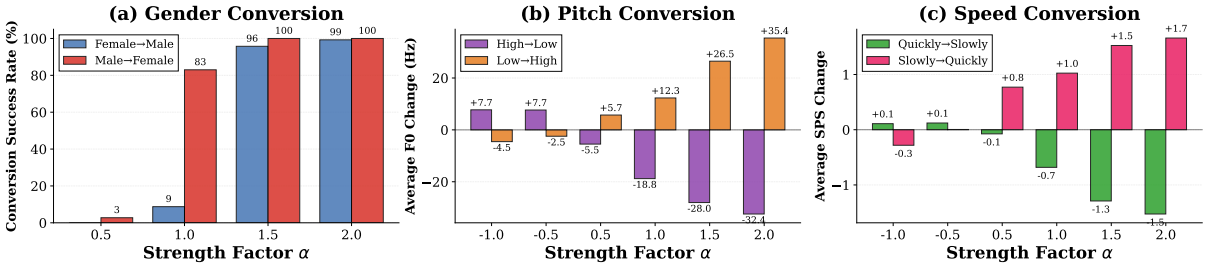


Figure 7: Inter-utterance style interpolation results when shifting all tokens. Results are comparable to attribute-only interpolation (Figure 3), demonstrating that manipulating only attribute tokens is sufficient for effective style control.

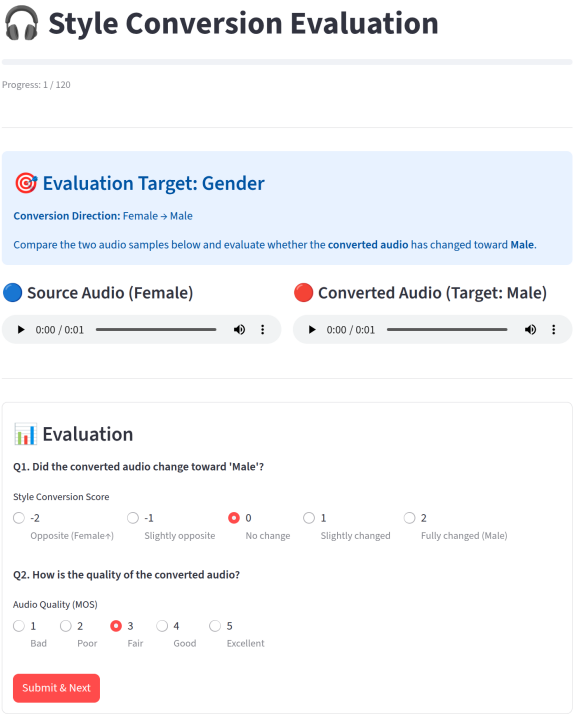


Figure 8: Evaluation interface for inter-utterance style interpolation. Participants compared source and converted audio, rating both the style conversion effectiveness (Q1) and audio quality (Q2).

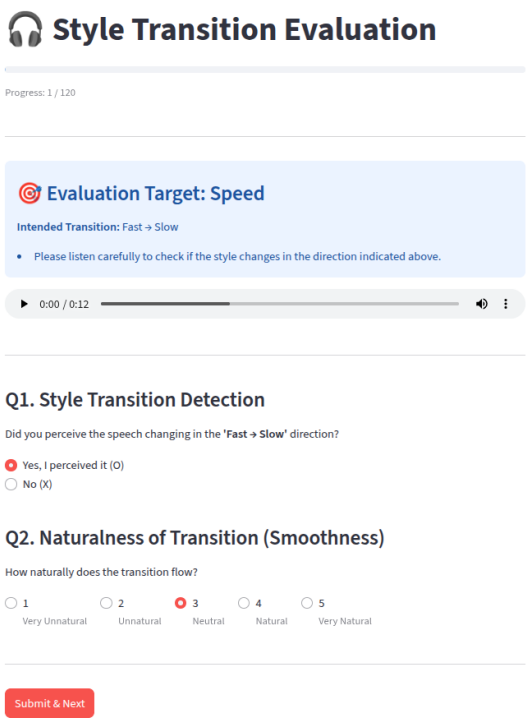


Figure 9: Evaluation interface for intra-utterance style transition. Participants evaluated whether they detected the intended style transition (Q1) and rated the naturalness of the transition (Q2).

to shift all token positions toward the target style representation.

We compare these two strategies by conducting experiments where all tokens are shifted. For attribute tokens, we vary $\alpha \in \{-1.0, -0.5, 0.5, 1.0, 1.5, 2.0\}$, while for non-attribute tokens, we apply a separate interpolation factor $\beta \in \{-0.5, -0.25, 0, 0.25, 0.5, 0.75, 1.0\}$ with increments of 0.25.

Figure 7 presents the results when shifting all tokens. Compared to the attribute-only results in Figure 3, both approaches achieve similar performance across all three attributes: gender conversion reaches a 99–100% success rate at $\alpha = 2.0$, pitch changes reach up to -32.4 Hz (High→Low) and $+35.4$ Hz (Low→High), and speed changes reach up to -1.5 SPS (Fast→Slow) and $+1.7$ SPS (Slow→Fast).

Table 5 summarizes the results at $\alpha = 2.0$ and $\beta = 1.0$. Compared to the attribute-only approach (Table 1), the full vector interpolation achieves comparable success rates and acoustic feature changes while maintaining similar speaker similarity. Since the full vector approach requires additional hyperparameter tuning for non-attribute tokens without providing clear benefits, we adopt the simpler attribute-only strategy in our main experiments.

Attribute	Direction	Success (%)	Δ Metric	SIM
Gender	Female → Male	99.3	–	–
	Male → Female	100	–	–
Pitch	High → Low	91.3	−32.4 Hz	0.79
	Low → High	93.5	+35.4 Hz	0.79
Speed	Fast → Slow	95.5	−1.5 SPS	0.85
	Slow → Fast	95.75	+1.7 SPS	0.85

Table 5: Full vector interpolation results at $\alpha = 2.0$ (attribute tokens) and $\beta = 1.0$ (non-attribute tokens).

C Quantitative Analysis of Early-Token Attention Bias

To quantitatively support the observation in Section 4.1.2 that cross-attention weights stabilize after the initial generation phase, we compute the variance of attention weights across style tokens at each audio generation position.

Specifically, let $\mathbf{a}_t \in \mathbb{R}^{|S|}$ denote the attention weight distribution over style tokens S at audio position t. We compute the variance of this distribution:

$$\mathrm{Var}(t) = \frac{1}{|S|} \sum_{s \in S} (a_{t,s} - \bar{a}_t)^2 \tag{5}$$

where $a_{t,s}$ is the attention weight from audio position t to style token s, and $\bar{a}_t$ is the mean attention weight at position t.

Figure 10 shows the attention variance across audio positions for multiple decoder layers. During the initial generation phase, the variance fluctuates significantly, indicating active attention weight updates as the model establishes style characteristics. After this phase, the variance stabilizes, confirming that fixed attention weights are assigned across all style tokens. This quantitative analysis supports our claim that the decoder actively queries style information during early generation and subsequently maintains a static attention pattern.

D Style Prompt Examples

Table 6 lists the style prompts used in our experiments for inter-utterance style interpolation and intra-utterance style variation. For each attribute, we define source and target prompts that differ only in the attribute-specific keywords (underlined).

For intra-utterance experiments, we use the same prompts but apply the direction vector manipulation at the transition point during autoregressive generation.

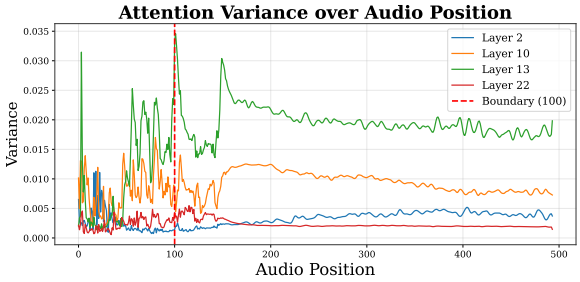


Figure 10: Attention weight variance across audio generation positions for different decoder layers. During early generation, variance fluctuates as the model actively updates attention weights to establish style characteristics. Afterward, variance becomes stable, indicating fixed attention patterns.

Attribute	Prompt
Gender	"A male voice speaks moderate at a medium pitch with monotone modulation and a clean quality." "A female voice speaks moderate at a medium pitch with monotone modulation and a clean quality."
Pitch	"A male voice speaks normally at a high pitch and a clean quality." "A male voice speaks normally at a low pitch and a clean quality."
Speed	"A male voice speaks quickly at a normal pitch and a clean quality." "A male voice speaks slowly at a normal pitch and a clean quality."

Table 6: Style prompts used in experiments. The attribute keywords (underlined) are the tokens where direction vectors are computed and applied.

E Algorithm for Intra-utterance Style Variation

Algorithm 1 provides the complete procedure for our intra-utterance style variation method, combining direction vector interpolation, a dual-decoder setup, KV-cache swap, and sliding window attention masking.

Algorithm 1 Intra-utterance Style Transition via Direction Vector and KV-cache Swap

Require: Text prompt P, source style text S, target style text T, source keyword, target keyword, transition point t^*, window size w, prefix steps k, strength α

Ensure: Generated audio with style transition

// Extract direction vector

Extract direction vector $\mathbf{d}$ from S and T at keyword positions

$n \leftarrow |P| + k$ // text length + prefix steps

// Initialize decoders with style embeddings

Initialize Decoder A with source style embeddings $\mathbf{E}^{(s)}$

Apply direction vector: $\mathbf{E}^{'} \leftarrow \mathbf{E}^{(s)} + \alpha \cdot \mathbf{d}$ at keyword position

Initialize Decoder B with target style embeddings $\mathbf{E}^{'}$

// Phase 1: Pre-compute target KV-cache

for $t = 1$ **to** n **do**

  Generate with Decoder B to build $\mathbf{K}_{1:n}^{(B)}, \mathbf{V}_{1:n}^{(B)}$

end for

// Phase 2: Generate until transition with sliding window

for $t = 1$ **to** t^* **do**

  Apply sliding window mask $M[i, j]$ with window size w ▷ Eq. 4

  Generate audio token with Decoder A

end for

// Phase 3: Swap at transition point

$\mathbf{K}_{1:n}^{(A)}, \mathbf{V}_{1:n}^{(A)} \leftarrow \mathbf{K}_{1:n}^{(B)}, \mathbf{V}_{1:n}^{(B)}$ ▷ Eq. 3

$\mathbf{E}^{(s)} \leftarrow \mathbf{E}^{'}$

// Phase 4: Continue generation with sliding window

for $t = t^* + 1$ **to** end **do**

  Apply sliding window mask $M[i, j]$ with window size w ▷ Eq. 4

  Generate audio token with Decoder A

end for

return Generated audio sequence