Overcoming State Inertia in Full-Duplex Spoken Language Models via Activation Steering

Cheng-Kuang Chang*
MIT CSAIL
chang168@mit.edu

Kai-Wei Chang*
MIT CSAIL
kwchang@mit.edu

Alexander H. Liu
MIT CSAIL
alexhliu@csail.mit.edu

James Glass
MIT CSAIL
glass@mit.edu

Abstract

Full-duplex spoken language models (FD-SLMs) process incoming speech while generating spoken responses, but do they reliably understand user speech while speaking? We find that these models can miss critical semantic information at the onset of user interruptions, even though their input channels remain active. To systematically evaluate this failure, we introduce the *Zero-Buffer Benchmark (ZBB)*, which places the critical semantic keyword at the beginning of an interrupting utterance without leading filler or an acoustic buffer. Across multiple FD-SLMs, interruptions reduce response correctness and Initial Word Occurrence Rate (IWOR), a behavioral indicator of initial-word recognition. To investigate these failures, we analyze FD-SLM hidden representations and identify *stream-specific predictive patterns*: representations preferentially predict the incoming user stream during listening and the model's own output stream during speaking. We characterize these relative predictive preferences as the *perceptive state* and the *generative state*, respectively. Following an interruption, the model can remain biased toward the generative state and fail to shift promptly toward the perceptive state. We term this delayed transition *state inertia*, which helps explain why early interruption content is missed despite continuous audio input. Finally, we construct a *perception vector* and apply it at interruption onset through training-free activation steering. By shifting the model toward the perceptive state, this intervention improves downstream interruption comprehension across multiple FD-SLMs. On PersonaPlex, correctness increases from 28% to 45%, and IWOR from 40% to 72%.

1 Introduction

Achieving human-level conversational fluency has long been a central goal in spoken dialogue systems [2, 18, 21]. Recently, *full-duplex spoken language models (FD-SLMs)* have attracted increasing attention for their ability to listen and speak simultaneously, moving beyond the rigid turn-by-turn interaction of conventional half-duplex spoken language models (HD-SLMs) [6, 5, 10, 39, 21, 24, 14, 41, 45]. These models typically process incoming user speech and their own speech as two concurrent streams [33, 13, 22, 2]. This design supports backchanneling, interruption handling, and fluid turn-taking [6, 24, 23, 11].

However, we find that when a user interrupts a speaking model, the model can miss the first words of the interruption, even though its input channel remains active. For example, when interrupted with

*Equal contribution.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

"Submarines fly in the sky, right?", Moshi responds as though the question were about airplanes, missing the word that determines the meaning of the utterance. This failure prompts us to ask: despite receiving both streams concurrently, *do full-duplex spoken language models really listen while speaking?*

To systematically evaluate this behavior, we introduce the **Zero-Buffer Benchmark (ZBB)**. ZBB places the critical semantic keyword at the beginning of an interrupting utterance, without leading filler or an acoustic buffer [8, 15]. This design tests whether a model captures semantic information from the onset of an interruption. We evaluate response correctness and Initial Word Occurrence Rate (IWOR), a behavioral indicator of initial-word recognition. Across multiple FD-SLMs, interruptions substantially reduce both metrics, showing that models can miss critical user information while speaking and consequently respond incorrectly.

To understand why these failures occur, we examine the predictive behavior encoded in FD-SLM hidden representations, drawing inspiration from *logit lens* [27, 4, 30]. We find *stream-specific predictive patterns*: hidden representations preferentially predict the incoming user stream during listening and the model's own output stream during speaking. We characterize these relative predictive preferences as the **perceptive state** and the **generative state**, respectively. Following an interruption, the model can remain biased toward the generative state and fail to shift promptly toward the perceptive state. We term this delayed transition **state inertia**, which helps explain why the model misses early interruption content despite continuously receiving user audio. Interestingly, "state inertia" resembles speech-induced suppression in human auditory processing, where speech production can suppress activity in the auditory cortex and increase auditory response latency [28, 20].

We further show that these states can be manipulated through *activation steering* [38, 49, 32]. We construct a *perception vector* from the difference between hidden representations in the perceptive and generative states and apply it at interruption onset to steer the model toward perception. This shift in internal state translates into improved interruption comprehension, reducing the performance degradation caused by interruptions on ZBB. On PersonaPlex [33], for example, correctness increases from 28% to 45%, and IWOR from 40% to 72%. These improvements support the connection between the model's internal state and its ability to understand incoming speech while speaking.

Our main contributions are summarized below.

- **Evaluating listening while speaking.** We show that FD-SLMs can miss critical user information while speaking and introduce ZBB to systematically evaluate comprehension at the onset of abrupt interruptions.
- **Internal state analysis and state inertia.** We identify perceptive and generative states through stream-specific predictive patterns in hidden representations. We uncover state inertia, a delayed transition toward perception that helps explain the observed interruption failures.
- **Training-free mitigation via activation steering.** We construct a perception vector that steers models toward the perceptive state, mitigating state inertia and substantially improving interruption comprehension across multiple FD-SLMs.

2 Related Work

Full-Duplex Spoken Language Models. Many existing spoken language models follow a half-duplex interaction pattern, processing input and output speech sequentially and relying on explicit turn-taking boundaries between listening and speaking [16, 44, 47]. This rigid interaction pattern can make conversations feel unnatural, especially in scenarios involving interruptions, backchannels, or overlapping speech [34]. In contrast, full-duplex spoken language models (FD-SLMs) support real-time bidirectional speech interaction, allowing the model to continuously perceive user audio while generating speech responses [2, 39, 48]. This capability enables more natural conversational behaviors, including backchanneling, interruption handling, and overlapping speech [24]. Motivated by these advantages, recent work has developed several full-duplex systems, including open-source models such as Moshi [13], PersonaPlex [33], and Raon-SpeechChat [22]. While these systems demonstrate the promise of full-duplex interaction, the internal mechanisms by which they coordinate simultaneous listening and speaking remain underexplored.

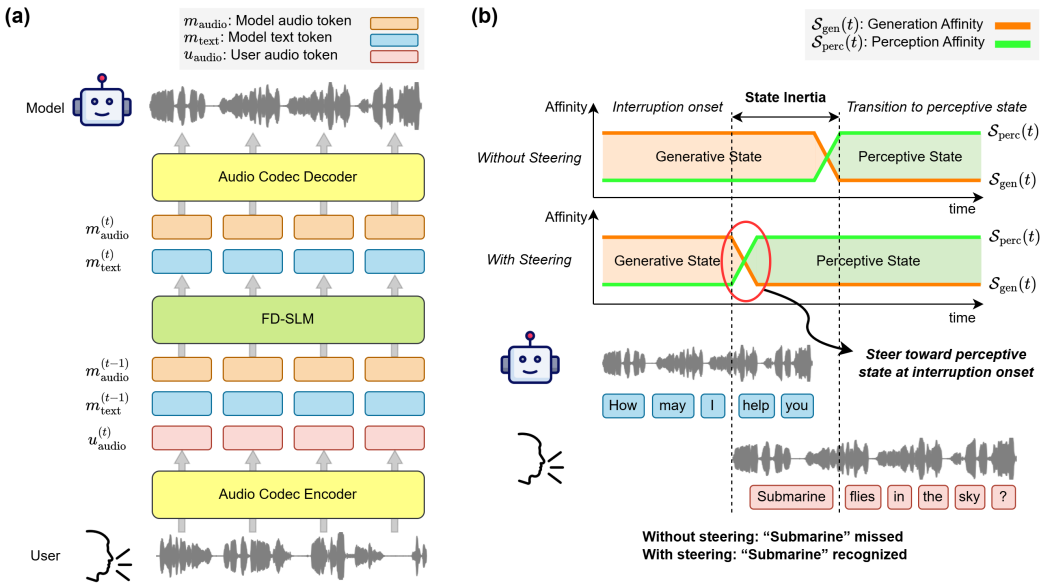

Figure 1: **Overview of state inertia and activation steering.** (a) FD-SLMs process concurrent user and model streams, conditioning on incoming user audio and previous model output tokens to generate text and audio tokens. (b) FD-SLMs coordinate speaking and listening by modulating between generative and perceptive states, tracked by generation and perception affinity. During abrupt interruptions, the model can remain biased toward the generative state before transitioning to the perceptive state, causing early user input to be missed. Injecting a perception vector at interruption onset accelerates this transition and improves interruption handling.

FD-SLMs Benchmarks. Existing benchmarks for FD-SLMs [23, 25, 29, 6] primarily assess macroscopic conversational properties. These include turn-taking dynamics, such as properly taking or yielding the floor; end-to-end response latency; overall instruction following; and full-duplex-specific behaviors such as backchanneling. However, these benchmarks largely overlook a critical fine-grained capability: whether the model accurately recognizes user input immediately following an abrupt interruption. This distinction is important because a model may eventually recover and produce a plausible response while still missing information at the beginning of the interrupting utterance. In this work, we assess this moment-level listening ability, which we discuss in Section 4.

Activation Steering. Activation steering modifies model behavior at inference time by injecting steering vectors into hidden states, often using mean-activation differences between contrasting concepts or behaviors [49, 38, 32]. Prior work has used steering to control text-generation behavior, such as instruction following, persona modification, vulnerability analysis, and representation probing [35, 7, 40, 1]. We instead apply activation steering to FD-SLMs, using it to steer hidden representations toward processing user input and improve immediate interruption handling.

3 Internal Mechanism of Full-Duplex SLMs

3.1 Full-duplex Spoken Language Model

As shown in Figure 1, Full-Duplex Spoken Language Models (FD-SLMs) process two concurrent speech streams: a *user stream* and a *model stream*. An audio codec discretizes the continuous speech signals into audio tokens, allowing the interaction to be represented as a sequence of timesteps [12, 46]. At each timestep t, the FD-SLM conditions on the incoming user audio tokens and its previously generated model tokens, and then produces the next model response. Practically, recent FD-SLMs first generate text tokens as a semantically rich *intermediate representation*, which then guides the generation of the corresponding speech [5, 13, 33, 22].

Formally, at timestep t, let $u^{(t)}_{\text{audio}}$ denote the user input audio tokens, and let $m^{(t)}_{\text{audio}}$ and $m^{(t)}_{\text{text}}$ denote the model output audio and text tokens, respectively. Let M_θ denote an FD-SLM parameterized by θ. At each timestep, M_θ generates the model output tokens $m^{(t)}_{\text{text}}$ and $m^{(t)}_{\text{audio}}$ conditioned on the current user input audio tokens $u^{(t)}_{\text{audio}}$, the model's previous audio and text tokens, and the preceding dialogue context $c^{(t)}$:

$$\left(m^{(t)}_{\text{audio}}, m^{(t)}_{\text{text}}\right) \sim M_\theta\left(\cdot \mid u^{(t)}_{\text{audio}}, m^{(t-1)}_{\text{audio}}, m^{(t-1)}_{\text{text}}, c^{(t)}\right), \tag{1}$$

where $c^{(t)}$ summarizes the dialogue history before timestep t.

Throughout the paper, we use a timestep as the minimal unit of processing rather than an individual token. Unlike text-only LLMs, FD-SLMs may contain multiple tokens at each timestep across parallel streams, making timesteps a more consistent unit for our analysis [13, 2, 9, 42].

3.2 Logit Lens

Unlike text-only LLMs or half-duplex SLMs, FD-SLMs must continuously coordinate listening to the user with generation of their own speech. However, how this coordination is represented internally remains poorly understood. To analyze this internal behavior, we use the *logit lens* [27, 4], which projects hidden representations from intermediate layers into the vocabulary space, allowing us to inspect how token-level predictions evolve across model depth.

Let $h^{(t)} \in \mathbb{R}^d$ denote the hidden representation at the selected layer and timestep t, and let $W_{\text{unembed}} \in \mathbb{R}^{|V|\times d}$ be the unembedding matrix, where V denotes the token vocabulary. For any target token $y \in V$, we define its projected probability under the hidden representation as

$$P(y \mid h^{(t)}) = \frac{\exp(w_y^\top h^{(t)})}{\sum_{v\in V}\exp(w_v^\top h^{(t)})}, \tag{2}$$

where $w_y^\top$ and $w_v^\top$ are the rows of W_{unembed} corresponding to tokens y and v, respectively.

At each timestep t, we then decode the most likely token under this projected distribution:

$$y^{(t)}_{\text{decode}} = \arg\max_{y\in V} P(y \mid h^{(t)}). \tag{3}$$

To understand how the model's internal behavior differs between listening and speaking, we construct a dataset for turn-by-turn interactions, where the model first listens to the user's speech and then speaks to respond. We conduct logit-lens analysis on PersonaPlex [33] to qualitatively compare hidden-representation predictions between the listening and speaking segments. Further details of the dataset construction are provided in Appendix A.1.

Finding 1

*FD-SLM hidden representations exhibit **stream-specific predictive focus**: during listening, they preferentially predict the incoming user stream, whereas during speaking, they preferentially predict the output model stream.*

Table 1 illustrates the predictive behavior on the user query "Can you compare renewable energy sources and explain their pros and cons in daily use?" While the user is speaking, the model stays silent because it is listening. Even so, logit-lens decoding of its intermediate layers anticipates the upcoming user words rather than the model's own output: after hearing "explain," intermediate layers decode tokens such as "why" and "how"; after hearing "their," they decode tokens such as "own" and "pro"; and subsequent predictions align with "and" and "cons." During model speaking, in contrast, the decoded tokens track the model's own output stream. Complete layer-wise decoding examples for both segments, together with additional decoded samples, are provided in Appendix E.

3.3 Generative and Perceptive State

The qualitative observation using logit lens suggests that hidden representations exhibit stream-specific predictive focus: their predictions can be more aligned with either incoming user input or

Table 1: Examples of logit-lens decoded predictions during a listening segment. Bold tokens indicate decoded predictions that match or anticipate the actual incoming user speech.

Current user token	explain	their	pros	and
Intermediate-layer decoded tokens	why, how, personal	own, **pro**	**and**	**con**, **cons**
Actual next user token	their	**pros**	**and**	**cons**

model output generation. Building on this observation, we quantify how this predictive focus shifts over time by defining two affinity scores: *generation affinity* and *perception affinity*.

Generation Affinity $\mathcal{S}_{\text{gen}}(t)$: Generation affinity $\mathcal{S}_{\text{gen}}(t)$ quantifies the extent to which the hidden representation $h^{(t)}$ supports generation of the output model stream. We define generation affinity as the mean projected probability assigned to the model output text token $m_{\text{text}}^{(t)}$ and audio token $m_{\text{audio}}^{(t)}$ conditioned on the current hidden representation $h^{(t)}$:

$$\mathcal{S}_{\text{gen}}(t) = \frac{1}{2}\left(P(m_{\text{audio}}^{(t)} \mid h^{(t)}) + P(m_{\text{text}}^{(t)} \mid h^{(t)})\right). \tag{4}$$

A high $\mathcal{S}_{\text{gen}}(t)$ indicates that $h^{(t)}$ is strongly aligned with the model's own output generation, suggesting that the FD-SLM is in a *generative state*.

Perception Affinity $\mathcal{S}_{\text{perc}}(t)$: Perception affinity $\mathcal{S}_{\text{perc}}(t)$ quantifies the extent to which the hidden representation $h^{(t)}$ supports prediction of the incoming user stream. We define perception affinity as the projected probability assigned to the next incoming user audio token $u_{\text{audio}}^{(t+1)}$ conditioned on the current hidden representation $h^{(t)}$:

$$\mathcal{S}_{\text{perc}}(t) = P(u_{\text{audio}}^{(t+1)} \mid h^{(t)}). \tag{5}$$

A high $\mathcal{S}_{\text{perc}}(t)$ indicates that $h^{(t)}$ is strongly aligned with predicting the incoming user audio, suggesting that the FD-SLM is in a *perceptive state*.

We compute $\mathcal{S}_{\text{gen}}(t)$ and $\mathcal{S}_{\text{perc}}(t)$ on the 100 examples from the turn-by-turn interaction dataset. For audio-token probabilities, we use the first codec codebook, which primarily encodes semantic speech information, while later residual codebooks encode finer acoustic details [13, 46, 12].[2] We align all examples by setting $t = 0$ to the end of the user utterance and average the resulting score trajectories across examples. For demonstration, we show the results on PersonaPlex.

Finding 2

FD-SLMs coordinate speaking and listening by dynamically modulating between generative and perceptive states.

As shown in Figure 2, $\mathcal{S}_{\text{gen}}(t)$ rises after $t = 0$, indicating a transition into the generative state as the model prepares to respond. Conversely, Figure 3 shows that $\mathcal{S}_{\text{perc}}(t)$ remains high while the user is speaking ($t < 0$), indicating a perceptive state, and then rapidly decays after the user utterance ends. Together, these results show that FD-SLMs do not maintain generation and perception uniformly throughout the interaction; instead, they reconfigure their generative and perceptive states according to the conversational role they currently occupy.

We note that the final layers show a different pattern: $\mathcal{S}_{\text{perc}}(t)$ remains low while $\mathcal{S}_{\text{gen}}(t)$ remains high even during user-speaking segments. This is expected because the final layers are closest to the output distribution and must still produce model tokens at every timestep, which often correspond to silence while the user is speaking.

3.4 State Inertia

Real-world spoken conversations often involve overlapping speech, including interruptions and backchanneling. Prior work reports that overlap occurs in over 40% of conversational turns [26, 19],

[2] Using only the first audio codebook also avoids FD-SLM-specific timing offsets associated with later residual codebooks.

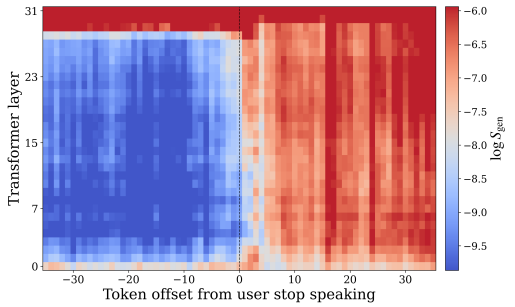

Figure 2: Generation affinity $\mathcal{S}_{\text{gen}}(t)$ across internal layers of PersonaPlex on the turn-by-turn interaction dataset. We align 100 examples at the end of the user utterance, with $t = 0$ marking this transition. Values are shown on a logarithmic scale.

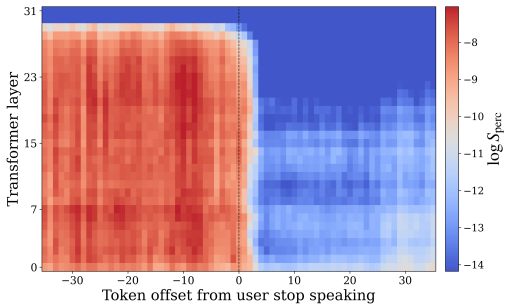

Figure 3: Perception affinity $\mathcal{S}_{\text{perc}}(t)$ across internal layers of PersonaPlex on the turn-by-turn interaction dataset. We align 100 examples at the end of the user utterance, with $t = 0$ marking this transition. Values are shown on a logarithmic scale.

making overlap handling an important capability for FD-SLMs. Unlike half-duplex systems, FD-SLMs are designed to listen while speaking; this simultaneous listening-and-speaking capability is a central motivation for full-duplex speech modeling.

In this work, we focus on user interruption as a representative and practically important form of speech overlapping. During an interruption, the user begins speaking while the model is still generating, and the model must quickly shift attention to the new input, yield the floor when appropriate, and respond to the updated conversational context. This scenario commonly arises in spoken assistant settings, where users may interrupt system speech to correct an error, redirect the dialogue, or provide input before the system finishes speaking [36, 31].

We compare how the generation and perception affinities, $\mathcal{S}_{\text{gen}}(t)$ and $\mathcal{S}_{\text{perc}}(t)$, evolve under two conditions: *interruption* and *no-interruption*. In the *interruption* condition, we first present a *speech-inducing prompt*: an open-ended question designed to place the model in a generative state. We then interrupt the model using a user utterance from the dataset introduced in the previous section. In the *no-interruption* condition, we present the same user utterance without first prompting the model to produce a substantive response. Detailed dataset construction is presented in Appendix A.2 For demonstration, we present an analysis using PersonaPlex as a representative example.

Finding 3

*The model exhibits **state inertia**: a tendency to remain in its prior state even when the conversational context requires an immediate transition.*

As shown in Figures 4 and 5, $\mathcal{S}_{\text{perc}}(t)$ remains low immediately after abrupt user input in the *interruption* condition compared with the *no-interruption* condition. This indicates that the model does not immediately transition out of the prompt-induced generative state. In this example, $\mathcal{S}_{\text{perc}}(t)$ takes approximately 7–8 timesteps, corresponding to about 0.6 seconds, to recover to the perceptive state. In contrast, under the *no-interruption* condition, the model transitions into the perceptive state almost immediately when the user begins speaking. We observe a similar delay in the generative-state transition, as shown in Appendix C. We refer to this delayed internal transition as **state inertia**.

4 Zero-Buffer Benchmark (ZBB)

A question naturally arises: whether state inertia, the delayed transition into the perceptive state, reduces the model's ability to perceive and understand user interruptions? To systematically quantify its impact on dialogue comprehension, we introduce the *Zero-Buffer Benchmark* (ZBB), which evaluates whether FD-SLMs can immediately understand user input when an interruption occurs. The key design principle is to place the critical semantic content at the very onset of the interrupting utterance, with no leading filler or acoustic buffer, so that the model must perceive core meaning exactly when state inertia is most likely to disrupt perception.

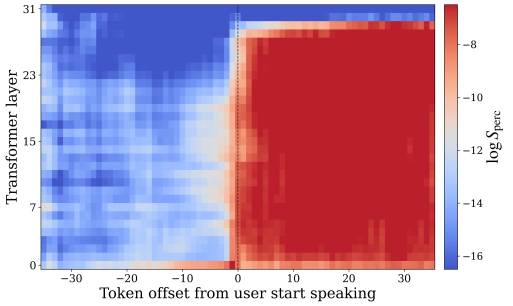

Figure 4: Perception affinity $\mathcal{S}_{\text{perc}}(t)$ in the *no-interruption* condition. The model transitions into the perceptive state immediately after the user begins speaking.

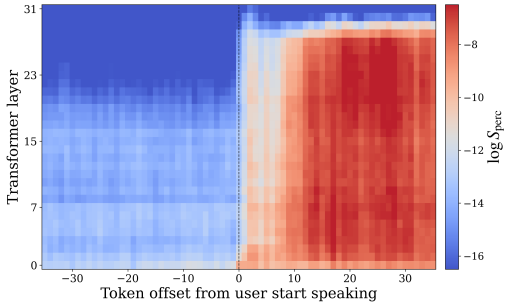

Figure 5: Perception affinity $\mathcal{S}_{\text{perc}}(t)$ in the *interruption* condition. The model transitions into the perceptive state after 7–8 timesteps, exhibiting state inertia.

Each ZBB example consists of a *speech-inducing prompt* followed by a *zero-buffer query*. The speech-inducing prompt is an open-ended question that places the model in a generative state; while the model is actively responding, we abruptly interrupt it with the zero-buffer query. Each zero-buffer query follows the template `<Subject>, <Description>, <Confirmation Request>` (e.g., *"Submarine flies in the clouds, right?"*), where the subject keyword is deliberately placed as the first word. Because the subject carries the information needed to judge the description, missing the onset of the interruption causes the model to lose the subject and often produce an incorrect or incoherent answer. The detail ZBB dataset creation and examples are provided in Appendix A.3.

For evaluation, we transcribe the generated audio and evaluate the following metrics with an LLM judge:

- **Correctness:** Whether the model answers the zero-buffer query correctly.
- **Initial Word Occurrence Rate (IWOR):** Whether the response explicitly mentions the initial semantic word of the zero-buffer query, or a direct synonym. IWOR provides a diagnostic measure of whether the model perceived the initial subject.

Evaluating several recent FD-SLMs on ZBB, we find that interruption substantially degrades both correctness and IWOR (Section 6.2), showing that state inertia has a measurable downstream impact on immediate interruption comprehension.

These results establish a behavioral deficit in comprehension at interruption onset. We next intervene on the model's internal predictive focus to examine its relationship with this deficit and mitigate the performance degradation caused by interruptions.

5 Activation Steering with Perception Vector

Can shifting the model toward the perceptive state improve its comprehension of user interruptions? To investigate this question, we apply activation steering [38] when the user begins speaking during model generation. This intervention allows us to examine whether manipulating the model's internal predictive focus improves downstream interruption comprehension.

We classify each timestep t as generation-dominant or perception-dominant using $\mathcal{S}_{\text{gen}}(t)$ and $\mathcal{S}_{\text{perc}}(t)$ computed at intermediate transformer layers. Specifically, we define $T_{\text{gen}} = \{t \mid \mathcal{S}_{\text{gen}}(t) \geq \Theta_{\text{gen}} \wedge \mathcal{S}_{\text{perc}}(t) < \Theta_{\text{perc}}\}$ and $T_{\text{perc}} = \{t \mid \mathcal{S}_{\text{perc}}(t) \geq \Theta_{\text{perc}} \wedge \mathcal{S}_{\text{gen}}(t) < \Theta_{\text{gen}}\}$, where Θ_{gen} and Θ_{perc} are predefined thresholds.

Following established representation engineering methods [38, 49, 32], we construct a *perception vector* as the difference between the mean hidden representations of perception-dominant and generation-dominant timesteps. Let $h^{(t)}$ denote the hidden representation at the selected steering layer and timestep t. We define the perception vector $\mu_{g \to p}$, which points from the generative state

toward the perceptive state, as

$$\mu_{g\to p} = \frac{1}{|T_{\text{perc}}|} \sum_{t \in T_{\text{perc}}} h^{(t)} - \frac{1}{|T_{\text{gen}}|} \sum_{t \in T_{\text{gen}}} h^{(t)}. \tag{6}$$

At inference time, we steer the model by adding the perception vector to the hidden representation at the selected steering layer, $\tilde{h}^{(t)} = h^{(t)} + \alpha \mu_{g\to p}$, where $\tilde{h}^{(t)}$ denotes the steered hidden representation and α controls the steering strength. In our ZBB experiments, steering is applied at the onset of the zero-buffer query, with the onset detected by an energy-based detector.

The geometry of the hidden representation space provides additional support for the perception vector. In Appendix D, we show that generation-dominant and perception-dominant timesteps are clearly separated under PCA projection. This separation suggests that the vector captures a meaningful transition direction rather than a noisy difference between overlapping distributions.

6 Experiments and Results on Zero-Buffer Benchmark

6.1 Setup

Evaluation conditions. We evaluate three advanced FD-SLMs spanning distinct architectural paradigms: PersonaPlex [33], Moshi [13], and Raon-SpeechChat [22]. For each model, we compare three conditions: *no interruption*, *interruption*, and *interruption with steering*. In the *interruption* condition, we first present a speech-inducing prompt and then abruptly interrupt the model with a zero-buffer query. In the *no-interruption* condition, we present the same zero-buffer query without first inducing substantive model speech. This condition represents the model's performance when no generative-to-perceptive transition is required. In the *interruption with steering* condition, we apply the perception vector at the onset of the zero-buffer query and measure whether it restores performance after interruption.

Perception vector construction. To construct the perception vector, we classify timesteps into T_{gen} and T_{perc} using the affinity scores defined in Section 3.3. For classification, we average $\mathcal{S}_{\text{gen}}(t)$ and $\mathcal{S}_{\text{perc}}(t)$ over layers 12–24 and apply the thresholds in Table 3. Unless otherwise stated, we use the steering layer, steering strength α, and steering span ΔT_{steer} specified in Table 3. Importantly, the conversations used to compute $\mu_{g\to p}$ are drawn from the turn-by-turn interaction dataset introduced in Section 3.2, and are disjoint from the ZBB evaluation set. Thus, the perception vector captures general state-level differences rather than information specific to the ZBB examples. Representative examples of these conversations are provided in Appendix A.

Steering schedule. At inference time, we apply the perception vector $\mu_{g\to p}$ starting at the onset of the zero-buffer query, denoted t_{int}. We detect t_{int} using an energy-based onset detector. Let $h^{(t)}$ denote the hidden representation at the selected steering layer and timestep t. To avoid steering the model throughout the entire interrupted utterance, we apply steering over a finite span ΔT_{steer} and linearly decay its magnitude to zero:

$$\tilde{h}^{(t)} = \begin{cases} h^{(t)} + \alpha \left(1 - \frac{t - t_{\text{int}}}{\Delta T_{\text{steer}}}\right) \mu_{g\to p}, & t_{\text{int}} \le t < t_{\text{int}} + \Delta T_{\text{steer}}, \\ h^{(t)}, & \text{otherwise}, \end{cases} \tag{7}$$

where $\tilde{h}^{(t)}$ denotes the steered hidden representation and α controls the steering strength.

6.2 ZBB Evaluation Results

As shown in Table 2, interruption causes a severe degradation in both correctness and IWOR across all three FD-SLMs. On PersonaPlex, for instance, correctness drops from 0.49 to 0.28 and IWOR from 0.74 to 0.40 when the query arrives as an interruption. The IWOR drop in particular indicates that the model often fails to perceive the initial subject of the interrupting utterance, showing that state inertia has a measurable downstream impact on immediate interruption comprehension.

Notably, activation steering improves both correctness and IWOR across all evaluated models. For PersonaPlex and Moshi, the perception vector raises response correctness and restores most of the interruption-induced IWOR drop (94% and 92%, respectively). For Raon-SpeechChat, steering improves both metrics as well, though absolute correctness remains low.

Table 2: FD-SLMs performance on ZBB. Uncertainties denote one standard error; parentheses show the percentage of the interruption-induced drop recovered by steering.

Model	Scenario	Correctness	IWOR
PersonaPlex	No Interrupt	0.49 ± 0.05	0.74 ± 0.04
	Interrupt	0.28 ± 0.04	0.40 ± 0.05
	Interrupt+Steer	$\mathbf{0.45 \pm 0.05}$ (81%)	$\mathbf{0.72 \pm 0.04}$ (94%)
Moshi	No Interrupt	0.43 ± 0.05	0.67 ± 0.05
	Interrupt	0.22 ± 0.04	0.29 ± 0.05
	Interrupt+Steer	$\mathbf{0.34 \pm 0.05}$ (57%)	$\mathbf{0.64 \pm 0.05}$ (92%)
Raon	No Interrupt	0.10 ± 0.03	0.29 ± 0.05
	Interrupt	0.03 ± 0.02	0.16 ± 0.04
	Interrupt+Steer	$\mathbf{0.17 \pm 0.03}$ (200%)	$\mathbf{0.24 \pm 0.04}$ (62%)

Table 3: Activation steering hyperparameters. Thresholds are reported in natural-log scale.

	PersonaPlex	Moshi	Raon
Layer	23	23	26
α	5.5	5.5	1.2
ΔT_{steer}	3	3	3
$\ln \Theta_{\text{gen}}$	-3.5	-3.5	-7.5
$\ln \Theta_{\text{perc}}$	-3.9	-3.9	-9.5

We further show qualitatively that activation steering reduces state inertia. We compare $\mathcal{S}_{\text{perc}}(t)$ around the onset of the zero-buffer query under the *interruption* and *interruption with steering* conditions in Figures 6 and 7, respectively. In the *interruption* condition, $\mathcal{S}_{\text{perc}}(t)$ remains low immediately after the zero-buffer query begins, indicating a delayed transition into the perceptive state. In contrast, under *interruption with steering*, $\mathcal{S}_{\text{perc}}(t)$ recovers immediately after the zero-buffer query onset. We provide an attention-based analysis in Appendix G, showing that steering increases attention to the first few interruption timesteps. Additional steering-parameter sweeps are provided in Appendix F.

We also evaluate steering on Full-Duplex Bench (FDB) [24] and confirm that steering does not degrade overall full-duplex dialogue performance. Results and discussion are provided in Appendix H.

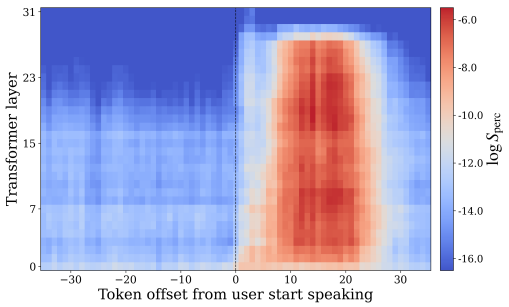

Figure 6: Perception affinity $\mathcal{S}_{\text{perc}}(t)$ in the *interruption* condition. Without steering, perception affinity takes approximately 7–8 timesteps to recover after interruption, indicating state inertia.

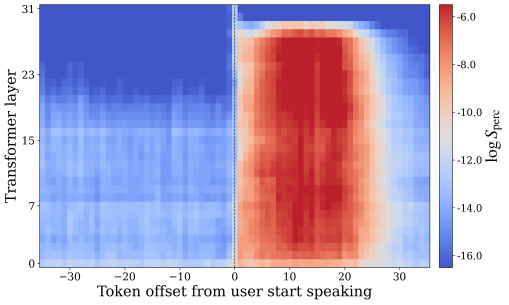

Figure 7: Perception affinity $\mathcal{S}_{\text{perc}}(t)$ in the *interruption with steering* condition. With activation steering, perception affinity recovers immediately after interruption, indicating a faster transition toward the perceptive state.

7 Limitations

Our work has several limitations. First, our evaluation covers a limited set of open-source FD-SLMs, and ZBB focuses specifically on comprehension at the onset of abrupt interruptions. The extent to which our findings generalize to other models and broader conversational settings remains to be established. Second, our logit-lens-based affinity scores are diagnostic approximations and can be noisy for individual examples. The perceptive and generative states characterize relative predictive preferences rather than provide a complete account of the mechanisms underlying listening–speaking coordination. Finally, our steering intervention relies on detecting interruption onset. We use an energy-based onset detector, whereas deployment in noisy or multi-speaker settings may require more robust detection. We discuss false-trigger sensitivity in Appendix I.

8 Conclusion

We investigate whether FD-SLMs reliably understand incoming speech while speaking and how their internal predictive behavior relates to interruption comprehension. Through the Zero-Buffer Benchmark (ZBB), we show that models can miss critical semantic information at interruption onset despite continuously receiving user audio. Our hidden-representation analysis reveals stream-specific predictive preferences, which we characterize as generative and perceptive states. We identify *state inertia*, a delayed shift toward the perceptive state following an interruption that helps explain failures to capture early user input. Beyond observing these dynamics, we use activation steering with a perception vector to influence the model's internal state. The resulting improvements in correctness and IWOR provide further evidence connecting internal predictive focus to interruption comprehension. Together, our findings show that concurrent audio processing does not by itself ensure reliable listening while speaking, and highlight internal-state dynamics as a useful perspective for understanding and influencing full-duplex conversational behavior.

References

[1] Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes, 2017. URL `https://openreview.net/forum?id=ryF7rTqgl`.

[2] Siddhant Arora, Kai-Wei Chang, Chung-Ming Chien, Yifan Peng, Haibin Wu, Yossi Adi, Emmanuel Dupoux, Hung-yi Lee, Karen Livescu, and Shinji Watanabe. On the landscape of spoken language models: A comprehensive survey. *Transactions on Machine Learning Research*, 2025.

[3] Joshua Ball. Voice activity detection (vad) in noisy environments. *arXiv preprint arXiv:2312.05815*, 2023.

[4] Nora Belrose, Igor Ostrovsky, Lev McKinney, Zach Furman, Logan Smith, Danny Halawi, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*, 2023.

[5] Kai-Wei Chang, Wei-Chih Chen, En-Pei Hu, Hung-yi Lee, and James Glass. TiCo: Time-controllable training for spoken dialogue models. *arXiv preprint arXiv:2603.22267*, 2026.

[6] Kai-Wei Chang, En-Pei Hu, Chun-Yi Kuan, Wenze Ren, Wei-Chih Chen, Guan-Ting Lin, Yu Tsao, Shao-Hua Sun, Hung-yi Lee, and James Glass. Game-time: Evaluating temporal dynamics in spoken language models. In *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 16302–16306. IEEE, 2026.

[7] Runjin Chen, Andy Arditi, Henry Sleight, Owain Evans, and Jack Lindsey. Persona vectors: Monitoring and controlling character traits in language models. *arXiv preprint arXiv:2507.21509*, 2025.

[8] Herbert H. Clark and Jean E. Fox Tree. Using uh and um in spontaneous speaking. *Cognition*, 84(1):73–111, 2002. ISSN 0010-0277. doi: https://doi.org/10.1016/S0010-0277(02)00017-3. URL `https://www.sciencedirect.com/science/article/pii/S0010027702000173`.

[9] Jade Copet, Felix Kreuk, Itai Gat, Tal Remez, David Kant, Gabriel Synnaeve, Yossi Adi, and Alexandre Défossez. Simple and controllable music generation. *Advances in neural information processing systems*, 36:47704–47720, 2023.

[10] Wenqian Cui, Dianzhi Yu, Xiaoqi Jiao, Ziqiao Meng, Guangyan Zhang, Qichao Wang, Steven Y Guo, and Irwin King. Recent advances in speech language models: A survey. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13943–13970, 2025.

[11] Wenqian Cui, Lei Zhu, Xiaohui Li, Zhihan Guo, Haoli Bai, Lu Hou, and Irwin King. Think before you talk: Enhancing meaningful dialogue generation in full-duplex speech language models with planning-inspired text guidance. *arXiv preprint arXiv:2508.07375*, 2025.

[12] Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. High fidelity neural audio compression. *Transactions on Machine Learning Research*, 2023.

[13] Alexandre Défossez, Laurent Mazaré, Manu Orsini, Amélie Royer, Patrick Pérez, Hervé Jégou, Edouard Grave, and Neil Zeghidour. Moshi: a speech-text foundation model for real-time dialogue. *arXiv preprint arXiv:2410.00037*, 2024.

[14] Ding Ding, Zeqian Ju, Yichong Leng, Songxiang Liu, Tong Liu, Zeyu Shang, Kai Shen, Wei Song, Xu Tan, Heyi Tang, et al. Kimi-audio technical report. *arXiv preprint arXiv:2504.18425*, 2025.

[15] Emily Duvall, Aimee Robbins, Thomas Graham, and Scott Divett. Exploring filler words and their impact. *Schwa. Language & Linguistics*, 11:35–49, 2014.

[16] Qingkai Fang, Shoutao Guo, Yan Zhou, Zhengrui Ma, Shaolei Zhang, and Yang Feng. LLaMA-omni: Seamless speech interaction with large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=PYmrUQmMEw`.

[17] Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495, 2021.

[18] James Glass. Challenges for spoken dialogue systems. In *Proceedings of the 1999 IEEE ASRU Workshop*, volume 696. MIT Laboratory for Computer Science Cambridge, 1999.

[19] Mattias Heldner and Jens Edlund. Pauses, gaps and overlaps in conversations. *Journal of Phonetics*, 38(4):555–568, 2010.

[20] John F Houde, Srikantan S Nagarajan, Kensuke Sekihara, and Michael M Merzenich. Modulation of the auditory cortex during speech: an meg study. *Journal of cognitive neuroscience*, 14(8):1125–1138, 2002.

[21] Shengpeng Ji, Yifu Chen, Minghui Fang, Jialong Zuo, Jingyu Lu, Hanting Wang, Ziyue Jiang, Long Zhou, Shujie Liu, Xize Cheng, et al. Wavchat: A survey of spoken dialogue models. *arXiv preprint arXiv:2411.13577*, 2024.

[22] Krafton. Raon-speech technical report, 2026.

[23] Guan-Ting Lin, Shih-Yun Shan Kuan, Jiatong Shi, Kai-Wei Chang, Siddhant Arora, Shinji Watanabe, and Hung-yi Lee. Full-duplex-bench-v2: A multi-turn evaluation framework for duplex dialogue systems with an automated examiner. *arXiv preprint arXiv:2510.07838*, 2025.

[24] Guan-Ting Lin, Jiachen Lian, Tingle Li, Qirui Wang, Gopala Anumanchipalli, Alexander H Liu, and Hung-yi Lee. Full-duplex-bench: A benchmark to evaluate full-duplex spoken dialogue models on turn-taking capabilities. *arXiv preprint arXiv:2503.04721*, 2025.

[25] Guan-Ting Lin, Chen Chen, Zhehuai Chen, and Hung-yi Lee. Full-duplex-bench-v3: Benchmarking tool use for full-duplex voice agents under real-world disfluency. *arXiv preprint arXiv:2604.04847*, 2026.

[26] Guan-Ting Lin, Shih-Yun Shan Kuan, Qirui Wang, Jiachen Lian, Tingle Li, Shinji Watanabe, and Hung-yi Lee. Full-duplex-bench v1. 5: Evaluating overlap handling for full-duplex speech models. In *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 19447–19451. IEEE, 2026.

[27] nostalgebraist. interpreting GPT: the logit lens, 2020. URL `https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens`.

[28] Jussi Numminen, Riitta Salmelin, and Riitta Hari. Subject's own speech reduces reactivity of the human auditory cortex. *Neuroscience Letters*, 265(2):119–122, 1999. ISSN 0304-3940. doi: https://doi.org/10.1016/S0304-3940(99)00218-9. URL `https://www.sciencedirect.com/science/article/pii/S0304394099002189`.

[29] Yizhou Peng, Yi-Wen Chao, Dianwen Ng, Yukun Ma, Chongjia Ni, Bin Ma, and Eng Siong Chng. Fd-bench: A full-duplex benchmarking pipeline designed for full duplex spoken dialogue systems. In *Proc. Interspeech 2025*, pages 176–180, 2025.

[30] Daking Rai, Yilun Zhou, Shi Feng, Abulhair Saparov, and Ziyu Yao. A practical review of mechanistic interpretability for transformer-based language models. *arXiv preprint arXiv:2407.02646*, 2024.

[31] Antoine Raux. Flexible turn-taking for spoken dialog systems. *Language Technologies Institute, CMU Dec*, 12, 2008.

[32] Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15504–15522, 2024.

[33] Rajarshi Roy, Jonathan Raiman, Sang-gil Lee, Teodor-Dumitru Ene, Robert Kirby, Sungwon Kim, Jaehyeon Kim, and Bryan Catanzaro. Personaplex: Voice and role control for full duplex conversational speech models. *arXiv preprint arXiv:2602.06053*, 2026.

[34] Gabriel Skantze. Turn-taking in conversational systems and human-robot interaction: a review. *Computer Speech & Language*, 67:101178, 2021.

[35] Alessandro Stolfo, Vidhisha Balachandran, Safoora Yousefi, Eric Horvitz, and Besmira Nushi. Improving instruction-following in language models through activation steering. In *The Thirteenth International Conference on Learning Representations*, 2024.

[36] Nikko Ström and Stephanie Seneff. Intelligent barge-in in conversational systems. In *INTERSPEECH*, pages 652–655, 2000.

[37] Ian Tenney, Dipanjan Das, and Ellie Pavlick. Bert rediscovers the classical nlp pipeline. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 4593–4601, 2019.

[38] Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*, 2023.

[39] Bandhav Veluri, Benjamin N Peloquin, Bokai Yu, Hongyu Gong, and Shyamnath Gollakota. Beyond turn-based interfaces: Synchronous llms as full-duplex dialogue agents. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21390–21402, 2024.

[40] Haoran Wang and Kai Shu. Trojan activation attack: Red-teaming large language models using steering vectors for safety-alignment. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 2347–2357, 2024.

[41] Boyong Wu, Chao Yan, Chen Hu, Cheng Yi, Chengli Feng, Fei Tian, Feiyu Shen, Gang Yu, Haoyang Zhang, Jingbei Li, et al. Step-audio 2 technical report. *arXiv preprint arXiv:2507.16632*, 2025.

[42] Haibin Wu, Ho-Lam Chung, Yi-Cheng Lin, Yuan-Kuei Wu, Xuanjun Chen, Yu-Chi Pai, Hsiu-Hsuan Wang, Kai-Wei Chang, Alex Liu, and Hung-yi Lee. Codec-superb: An in-depth analysis of sound codec models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 10330–10348, 2024.

[43] Kangxiang Xia, Bingshen Mu, Xian Shi, Jin Xu, and Lei Xie. Semantic-aware interruption detection in spoken dialogue systems: Benchmark, metric, and model. *arXiv preprint arXiv:2603.24144*, 2026.

[44] Zhifei Xie and Changqiao Wu. Mini-omni: Language models can hear, talk while thinking in streaming. *arXiv preprint arXiv:2408.16725*, 2024.

[45] Jin Xu, Zhifang Guo, Hangrui Hu, Yunfei Chu, Xiong Wang, Jinzheng He, Yuxuan Wang, Xian Shi, Ting He, Xinfa Zhu, et al. Qwen3-omni technical report. *arXiv preprint arXiv:2509.17765*, 2025.

[46] Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. Soundstream: An end-to-end neural audio codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:495–507, 2021.

[47] Aohan Zeng, Zhengxiao Du, Mingdao Liu, Kedong Wang, Shengmin Jiang, Lei Zhao, Yuxiao Dong, and Jie Tang. Glm-4-voice: Towards intelligent and human-like end-to-end spoken chatbot. *arXiv preprint arXiv:2412.02612*, 2024.

[48] Xinrong Zhang, Yingfa Chen, Shengding Hu, Xu Han, Zihang Xu, Yuanwei Xu, Weilin Zhao, Maosong Sun, and Zhiyuan Liu. Beyond the turn-based game: Enabling real-time conversations with duplex models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11543–11557, 2024.

[49] Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*, 2023.

A Dataset Details

A.1 Turn-by-turn interaction dataset

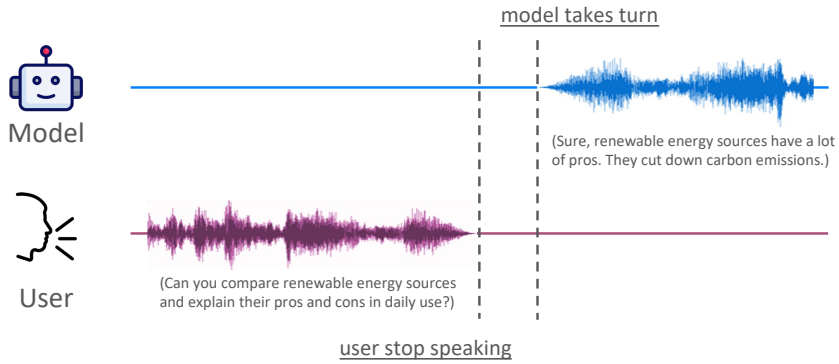

Figure 8: An example from the turn-by-turn interaction dataset used for logit-lens analysis and model-internal generation/perception affinity analysis.

The turn-by-turn interaction dataset consists of 100 *user queries* covering a diverse set of everyday conversational topics, each followed by a response window in which the model takes its turn to reply. We use this dataset for our logit-lens analysis, and to identify the generative and perceptive states by computing the generation and perception affinities.

We generate these user queries with a text-based LLM (Claude Opus 4.5) according to the following criteria: (1) the utterances should cover varied topics from daily conversation in order to increase diversity; (2) they should be open-ended, so that model responses are not biased toward a fixed answer format; and (3) after text-to-speech synthesis, they should correspond to approximately 15–20 seconds of speech, providing a sufficiently long listening segment for analysis. Example queries are shown below.

Example User query 1

Can you compare renewable energy sources and explain their pros and cons in daily use?

Example User query 2

My neighbor got this new puppy last week. Cutest little thing you've ever seen, but it barks all night long. I mean, non-stop. I haven't slept properly in days. I don't want to be rude about it, but I'm seriously considering saying something to her about the noise.

After generating the text queries, we synthesize them into speech using the `Dia2-2B` text-to-speech (TTS) model[3]. Because FD-SLMs operate on continuous audio input, each synthesized user utterance is followed by a 10-second silence segment, during which the model is allowed to respond. Thus, each audio input is approximately 25–30 seconds long: the first 15–20 seconds contain user speech, during which the model is expected to listen, and the final 10 seconds provide a response window for the model. The dataset contains 100 such examples.

A.2 Interruption and No-Interruption Conditions for Analyzing State Inertia

To analyze state inertia, we construct paired *no-interruption* and *interruption* conditions from the same user queries in Appendix A.1.

For the no-interruption condition, we present a user query on its own. The model is therefore not speaking when the user begins, yielding an ordinary turn-taking dialogue with no overlap. This setting is the same as in the turn-by-turn interaction dataset.

[3] `https://huggingface.co/nari-labs/Dia2-2B`

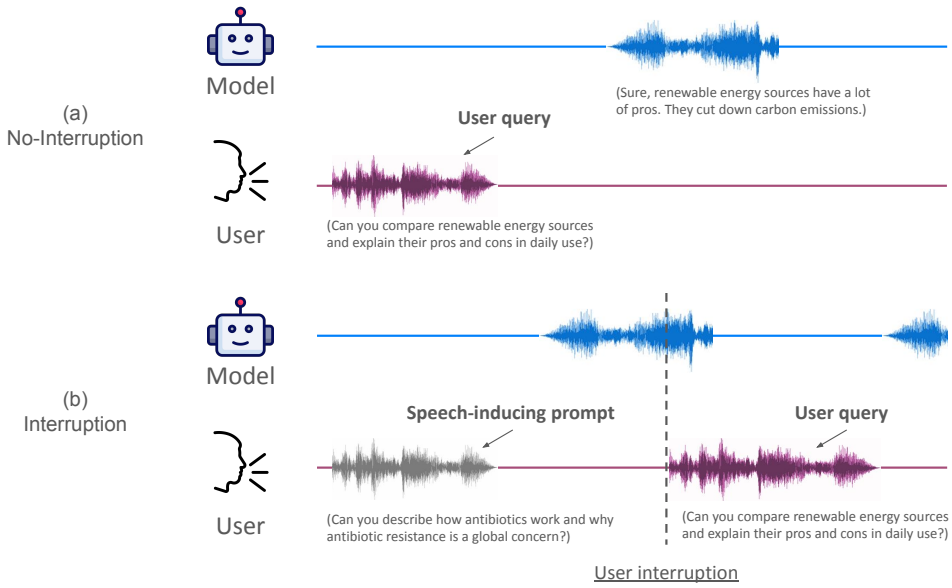

Figure 9: An example from the dataset for state inertia analysis, illustrating the paired (a) no-interruption and (b) interruption conditions. In the interruption condition, a speech-inducing prompt first places the model in a generative state, and a user utterance then interrupts its ongoing response; in the no-interruption condition, the same utterance is presented without a preceding prompt.

For the interruption condition, we first input a user *speech-inducing prompt*, which is an open-ended question designed to drive the model into a sustained generative state by eliciting a long response. These speech-inducing prompts are constructed according to the following criteria: (1) they should cover diverse topics to reduce topic bias; (2) they should involve relatively technical or explanatory content, so that the model is likely to produce a longer response; and (3) they do not need to be long, since their purpose is only to induce model-side speaking behavior. The speech-inducing prompts are generated using Claude Opus 4.5 and synthesized into speech using `Dia2-2B`.

An example speech-inducing prompt is shown below.

Example Speech-Inducing Prompt

Can you describe how antibiotics work and why antibiotic resistance is a global concern?

After receiving the speech-inducing prompt, the model begins generating a response; after 5 seconds, we abruptly interrupt it with the user query. This setup creates an interruption condition in which the model must transition from an ongoing generative state to a perceptive state.

A.3 Zero-Buffer Benchmark Dataset

As described in Section 4, the Zero-Buffer Benchmark (ZBB) contains two evaluation conditions: an interruption condition and a no-interruption condition. In the interruption condition, each example consists of a *speech-inducing prompt* followed by a *zero-buffer query*. In the no-interruption condition, the model receives the same zero-buffer query without first being induced into a sustained speaking state. This paired design allows us to measure how interruption affects both response correctness and initial-word recognition.

The speech-inducing prompts are constructed in the same way as in Appendix A.2. Each zero-buffer query follows the template

`<Subject>, <Description>, <Confirmation Request>.`

The subject appears as the first word of the query, so missing the onset of the interruption often removes the key information needed to answer correctly. To balance the dataset, we generate 50

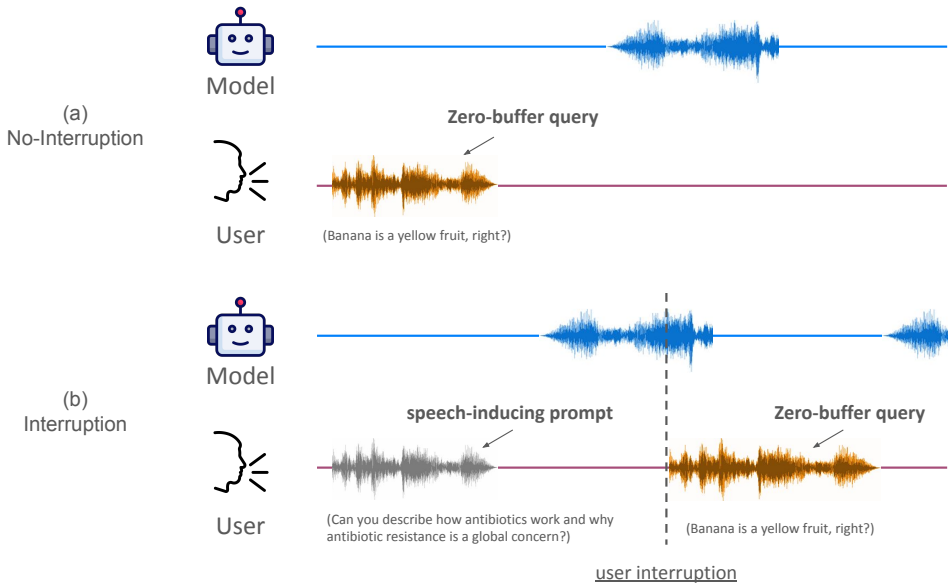

Figure 10: An example from the ZBB dataset, showing the paired (a) no-interruption and (b) interruption conditions. In the no-interruption condition, the zero-buffer query is presented on its own. In the interruption condition, a speech-inducing prompt is followed by a zero-buffer query that interrupts the model's ongoing response, testing whether the model can perceive the critical information at the onset of the interruption.

subjects. For each subject, we create one factually correct description and one factually incorrect description, resulting in 100 zero-buffer queries in total. The confirmation request is kept short, so that the first word remains the primary semantic cue at the onset of the interruption.

The subjects are chosen from common entities, objects, and animals, so that the expected answer is unambiguous and does not require specialized knowledge.

An example positive–negative pair is shown below.

Example Zero-Buffer Query Pair

Correct: *Banana is a yellow fruit, right?*
Incorrect: *Banana is a red fruit, right?*

Pairing the same subject with both a correct and an incorrect description helps control for subject-specific difficulty. In this way, differences in correctness are less likely to be explained by some subjects being inherently easier or harder to recognize.

The speech-inducing prompts and zero-buffer queries are synthesized into audio using the `Dia2-2B` text-to-speech model[4].

A.4 LLM-Based Evaluation for ZBB

We evaluate model responses using two metrics: correctness and Initial Word Occurrence Rate (IWOR). For both metrics, we first transcribe the model's generated speech into text using the ASR model `nvidia/parakeet-tdt-0.6b-v2`[5]. We then evaluate the transcription using GPT-4.1-mini with the prompts below.

For correctness, the evaluator determines whether the model gives a factually correct and direct answer to the interruption query.

[4] `https://huggingface.co/nari-labs/Dia2-2B`
[5] `https://huggingface.co/nvidia/parakeet-tdt-0.6b-v2`

CORRECTNESS_SYSTEM_PROMPT

```
Evaluate if the AI provided a FACTUALLY CORRECT and DIRECT answer to the
User's Interruption Question. Do NOT evaluate politeness, tone, or
conversational flow. Be strictly objective about accuracy.

Score 1: The response provides an accurate, specific, and logically correct
answer to the interruption question.
Score 0: The response is incorrect, evasive, or answers the wrong question.

Automatically assign 0 for any of these failures:
- Topic Bleed (Inertia): The AI continues answering or referencing the
  pre-interruption topic, failing to fully transition its logic to the
  new question.
- Factually Incorrect/Hallucination: The AI attempts to answer the new
  question but provides definitively wrong information.
- Evasion/Refusal: The AI dodges the question or gives a meaningless
  generic response.
- Vague Approximation: The answer lacks the specific detail required by
  the question.

Output ONLY JSON:
{"score": <0 or 1>, "reason": "<1-sentence explanation stating what the
correct answer should be and why the AI passed/failed>"}
```

For IWOR, the evaluator determines whether the model response explicitly mentions the subject entity appearing as the first word of the interruption query, or a direct synonym. This metric is designed to measure whether the model perceived the initial semantic keyword of the interruption.

FIRST_WORD_SYSTEM_PROMPT

```
Evaluate if the AI successfully heard the VERY FIRST WORD of the
User's Interruption Question. Do NOT infer context. Be strictly literal.

Score 1: The response EXPLICITLY names the entity/subject of the first
word (or a direct synonym).
Score 0: The response does NOT explicitly name the first word's subject.

Automatically assign 0 for any of these failures:
- Tail-End Catching: Reacts only to the end of the question, missing the
  first word.
- Pronoun Dodging: Uses "it/they/that/this" instead of explicitly naming
  the subject.
- Self-Referential: Answers with "I/my/me" because it missed the subject.
- Naked Answers: Just says "Yes/No/True/False" with no subject attached.
- Unrelated/Gibberish: Fails to address the first word entirely.

Output ONLY JSON:
{"score": <0 or 1>, "reason": "<1-sentence explanation identifying the
first word and whether it was explicitly said>"}
```

The final correctness score is the fraction of examples for which the evaluator assigns a score of 1 under the correctness rubric. The final IWOR score is the fraction of examples for which the evaluator assigns a score of 1 under the first-word rubric.

The following example illustrates the correctness evaluation.

Example of Correctness Evaluation

```
{
  "interruption_question": "Bicycle has four wheels, right?",
```

```
  "model_response": "Yeah, a bicycle has four wheels, that's right.",
  "gpt_score": 0,
  "gpt_reason": "The response incorrectly states that a bicycle has four wheels,
     whereas a bicycle actually has two wheels."
}
```


The following example illustrates the IWOR evaluation.

Example of IWOR Evaluation


```
{
  "interruption_question": "Bicycle has two wheels, right?",
  "model_response": "Yeah, that's right. A bike has two wheels.",
  "gpt_score": 1,
  "gpt_reason": "The first word 'Bicycle' is explicitly referred to as 'A bike'
    in the response."
}
```


Correctness and IWOR capture complementary aspects of interruption handling. Correctness measures whether the model answers the full interruption query accurately, whereas IWOR measures whether the model perceived the initial semantic keyword. A model may answer incorrectly even after recognizing the first word, or it may respond to the tail end of the question without explicitly recognizing the subject. We therefore report both metrics.

B Computational Resources

All experiments in this paper are conducted on NVIDIA L40S GPUs. Our experiments involve inference-time analysis and activation steering on open-source FD-SLMs, without model training or fine-tuning. Therefore, the compute requirements are modest compared with training-based approaches. The experiments can be run on any GPU with sufficient memory to host the evaluated models, including PersonaPlex, Moshi, and Raon-SpeechChat.

C Delayed Transition Out of the Generative State

In addition to the delayed transition into the perceptive state discussed in the main text, we also observe a delayed transition out of the generative state. Figure 11 and Figure 12 compare $\mathcal{S}_{\text{gen}}(t)$ under the *no-interruption* and *interruption* conditions, respectively. Under the *no-interruption* condition, generation affinity decreases shortly after the user begins speaking, indicating that the model exits the generative state relatively quickly. In contrast, under the *interruption* condition, $\mathcal{S}_{\text{gen}}(t)$ remains elevated for substantially longer after the user begins speaking, indicating that the model continues to occupy the generative state despite the change in conversational context. This provides complementary evidence for **state inertia**: the model exhibits a delayed internal transition not only into the perceptive state, but also out of the generative state.

D PCA of Hidden Representations

The perception vector $\mu_{g \to p}$ is computed as the difference between the mean hidden representations of perception-dominant and generation-dominant timesteps. This mean-difference direction is meaningful only if the two underlying representation distributions are sufficiently separated; if they heavily overlap, the resulting vector could instead reflect noise from weakly distinguishable distributions. To examine this possibility, we analyze the separability of these hidden representations using Principal Component Analysis (PCA).

As shown in Figure 13, generation-dominant and perception-dominant timesteps form clearly separated clusters in the PCA-projected hidden space across most layers. This separation supports the validity of the perception vector: it is not merely a noisy difference between overlapping distributions, but a direction aligned with a prominent structure in the model's hidden representations.

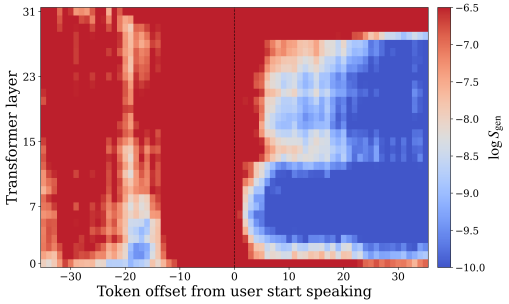

Figure 11: Generation affinity $\mathcal{S}_{\text{gen}}(t)$ in the *no-interruption* condition. The model exits the generative state soon after the user begins speaking, with recovery occurring after approximately 5 timesteps.

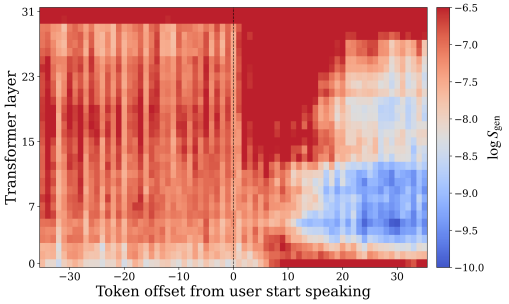

Figure 12: Generation affinity $\mathcal{S}_{\text{gen}}(t)$ in the *interruption* condition. The model remains in the generative state for approximately 20 timesteps after the user interrupts and begins speaking, corresponding to nearly 2 seconds.

The dominant separating component varies across depth. In lower layers, the two sets are primarily separated along the first principal component, whereas in deeper layers the separation becomes more apparent along the second principal component. One possible interpretation is that the dominant sources of variance change across layers: lower layers may emphasize surface-level or modality-specific structure, while deeper layers may allocate the leading principal component to content-related variation [17, 37], leaving state-related variation to appear in a secondary component. We treat this explanation as suggestive rather than conclusive.

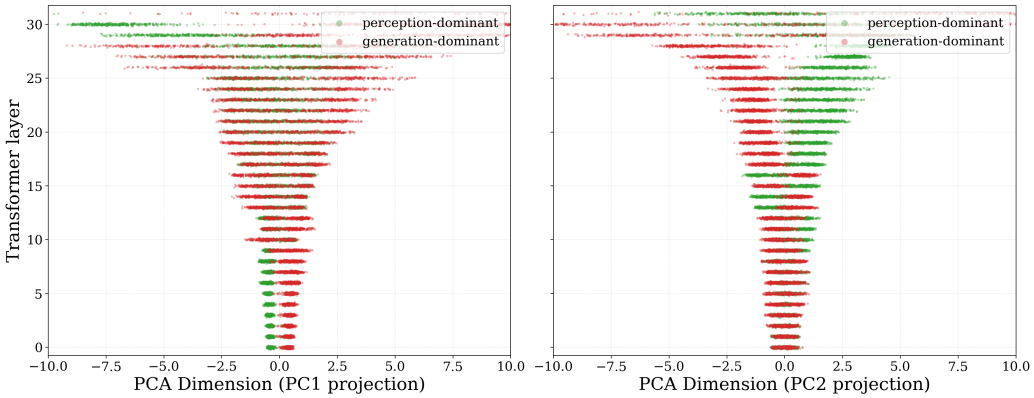

Figure 13: PCA projections of hidden representations from generation-dominant and perception-dominant timesteps across transformer layers. Generation-dominant and perception-dominant representations form separated clusters in the projected space. The separation is most visible along the first principal component in shallower layers (left) and along the second principal component in deeper layers (right).

E Decoding Hidden States with the Logit Lens

This appendix provides detailed qualitative examples from the turn-by-turn interaction dataset, complementing the analysis in Section 3.2. We visualize the top logit-lens prediction at each layer and timestep. For each hidden representation $h^{(t)}$, we project it into the vocabulary space using the same probability definition as in Section 3.2, and decode

$$y^{(t)}_{\text{decode}} = \arg\max_{y \in V} P(y \mid h^{(t)}). \tag{8}$$

In each heatmap, the text annotation in a cell shows $y^{(t)}_{\text{decode}}$, while the color indicates the projected probability assigned to the eventual model-side text token $m^{(t)}_{\text{text}}$.

Table 4: Examples of logit-lens decoded predictions during listening. Bold tokens match or anticipate the actual upcoming user-side token.

Current user token	Intermediate-layer decoded tokens	Actual next user token
explain	why, how, personal	their
their	own, **pro**	**pros**
pros	**and**	**and**
and	**con**, **cons**	**cons**

E.1 Logit-Lens Decoding During Listening

Figure 14 shows that, during listening, intermediate layers often predict continuations of the incoming user utterance rather than only the model-side output token. For example, when the user-side phrase is "their pros and cons," decoded tokens include "pro," "and," and "cons," which anticipate upcoming user-side content. The decoded tokens may also be semantically related to the ongoing utterance even when they do not exactly match the next token. For example, at the timestep corresponding to the input token "explain," the decoded tokens include "why," "how," and "personal," which are relevant continuations. We highlight several representative examples in Table 4. An additional layer-wise logit-lens decoding example is provided in Figure 15.

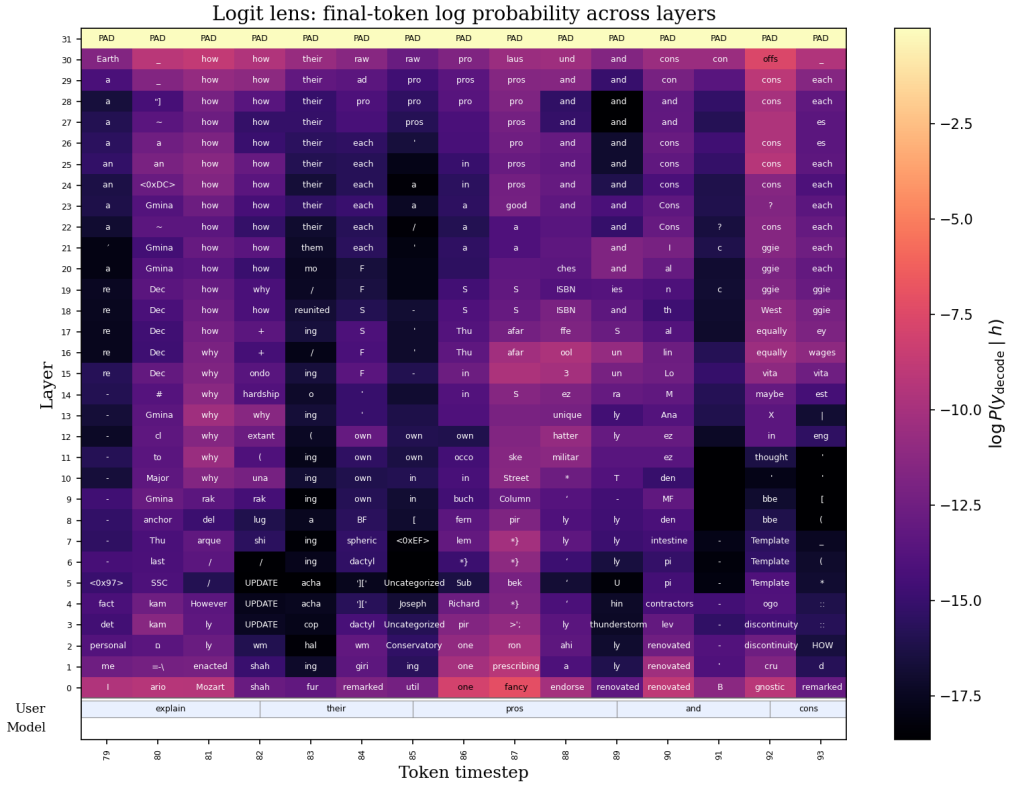

Figure 14: Logit-lens decoding of PersonaPlex hidden states during a listening segment. Intermediate layers often decode tokens related to the incoming user stream, even though the final model-side output remains mostly <PAD>. This suggests that the model internally tracks user-side content before converting this computation into a silent model-side output.

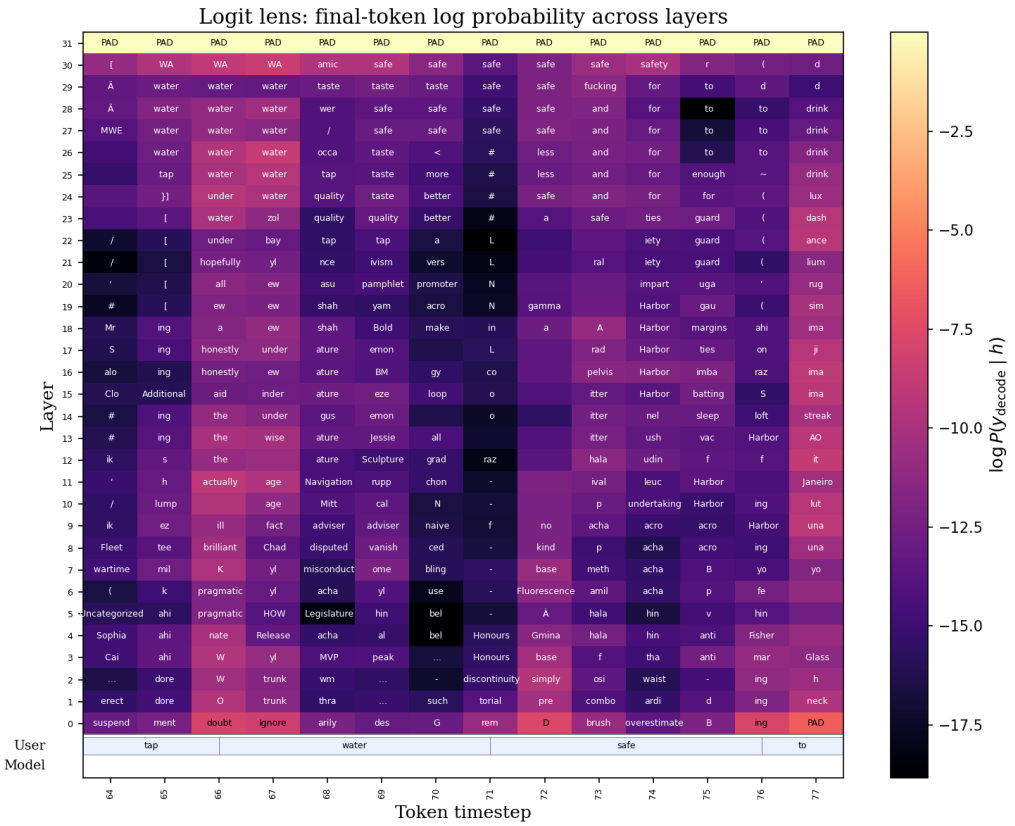

Figure 15: Additional logit-lens decoding example during a listening segment. The user input is "How does water treatment make tap water safe to drink in modern cities?" Intermediate layers decode tokens that anticipate or semantically track the incoming user stream: around "tap," decoded tokens include "water"; around "water," decoded tokens include "quality," "safe," and "tastes"; around "safe," decoded tokens include "to," "for," and "safety"; and around "to," decoded tokens include "drink." This provides further qualitative evidence that hidden states can track user-side continuations during listening.

E.2 Logit-Lens Decoding During Model Speech

Figure 16 shows the complementary pattern during model speech. Intermediate hidden states assign higher projected probability to model-side text tokens, and decoded tokens more directly follow the model output stream. Some timesteps still have lower model-text probability because recent FD-SLMs often distribute text-token and audio-token generation across different frames; during audio-generation frames, the model-side text token may be <PAD> or <EPAD>. An additional layer-wise logit-lens decoding example is provided in Figure 17.

Together, Figures 14 and 16 provide qualitative evidence for stream-specific predictive focus: hidden states tend to track the incoming user stream during listening and the model-side output stream during speaking. This supports the interpretation of $S_{\text{perc}}(t)$ and $S_{\text{gen}}(t)$ in Section 3.3 as indicators of perceptive and generative states, respectively.

F Steering Parameter Analysis

Steering layer and strength α. We investigate how the steering layer and steering strength α affect ZBB performance. We perform a grid search over candidate steering layers and values of α on PersonaPlex. As shown in Figure 18, steering is most effective at layer 23 across the tested

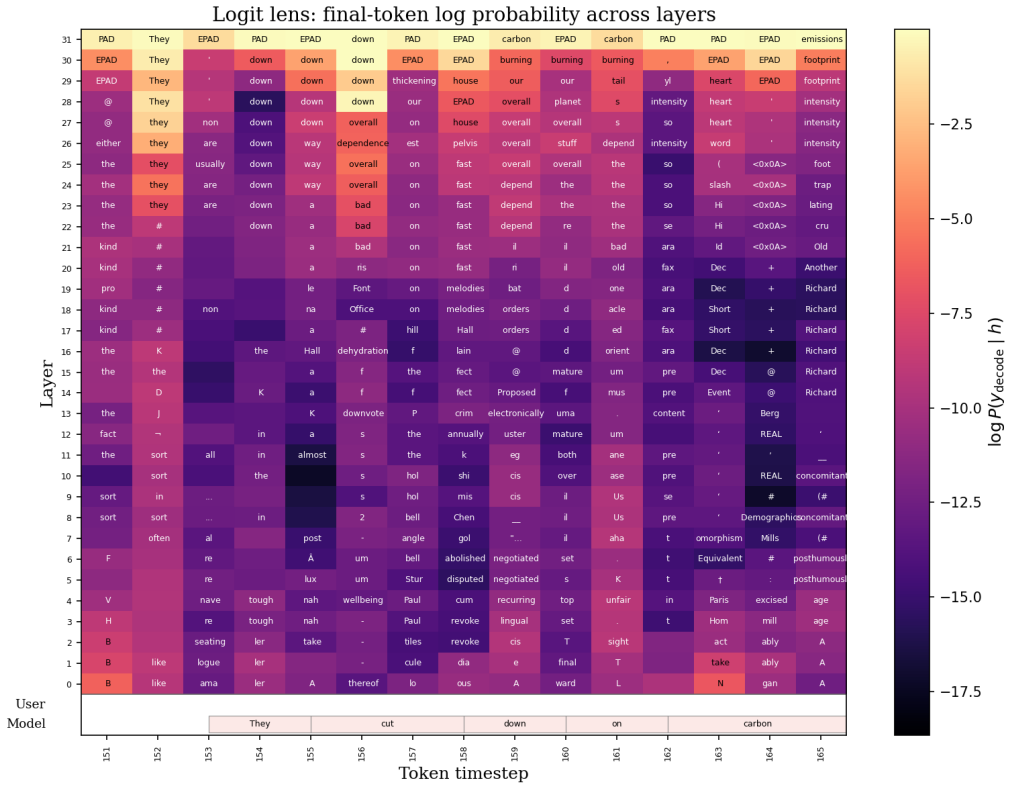

Figure 16: Logit-lens decoding of PersonaPlex hidden states during a model speaking segment. Compared with the listening segment in Figure 14, the speaking segment shows stronger alignment with the model-side output stream across a broader range of layers, consistent with a generative state.

values of α. The best configuration is achieved at $\alpha = 5.5$, where correctness reaches 0.45 and IWOR reaches 0.72.

Steering span ΔT_{steer}. We further investigate how the steering span affects ZBB performance. For this scan, we fix the steering layer to 23 and the steering strength to $\alpha = 5.5$. As shown in Figure 19, short steering spans already improve both correctness and IWOR over the interruption condition in Section 6.2, while a span of 3 timesteps achieves the best overall performance. Longer spans gradually reduce performance, suggesting that steering is most effective when applied briefly at the interruption onset rather than throughout the interrupted utterance.

G Attention Recovery After Steering

Given that activation steering improves both correctness and IWOR, we further examine whether it changes attention allocation after interruption. Specifically, we measure how strongly subsequent timesteps attend back to earlier timesteps in the interrupting user input.

We compute the average attention weight assigned to the input at timestep t by the subsequent n timesteps at the attention layer of interest. Let $w_j(t, \tau)$ denote the attention weight from the query at timestep τ to the key at timestep t in attention head j, and let $\mathcal{H}$ denote the set of attention heads in this layer. We define s_t as the average attention score assigned to timestep t over the next n timesteps, averaged across all attention heads:

$$s_t = \frac{1}{n|\mathcal{H}|} \sum_{\tau=t+1}^{t+n} \sum_{j \in \mathcal{H}} w_j(t, \tau). \tag{9}$$

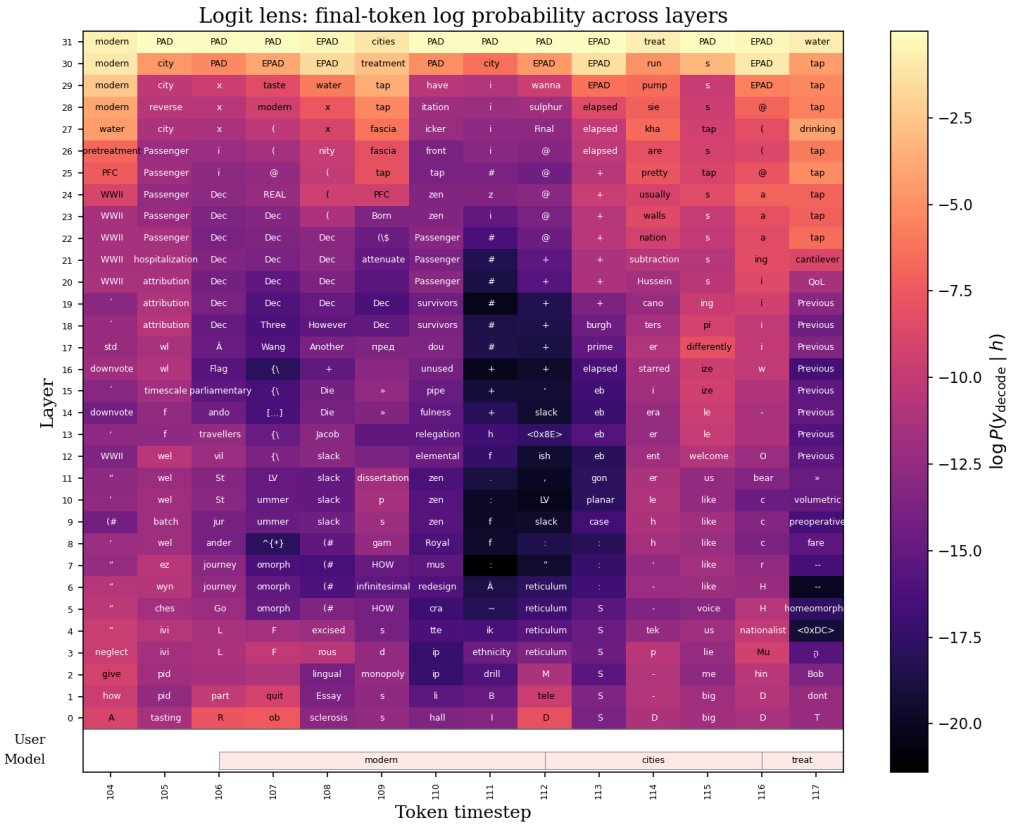

Figure 17: Additional logit-lens decoding example during a model speaking segment. This example corresponds to the model response beginning with "Modern cities treat water..." after the user query shown in Figure 15. The decoded tokens follow the model-side output stream, providing further qualitative evidence of generative-state alignment during speaking.

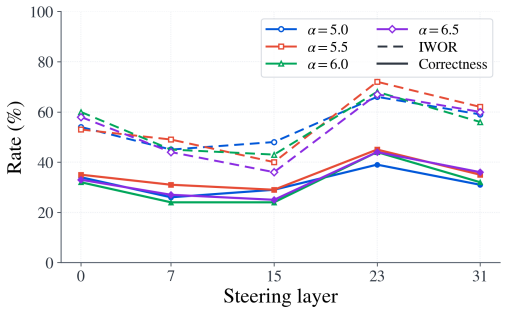

Figure 18: Correctness and IWOR across steering layers for different steering strengths α on PersonaPlex.

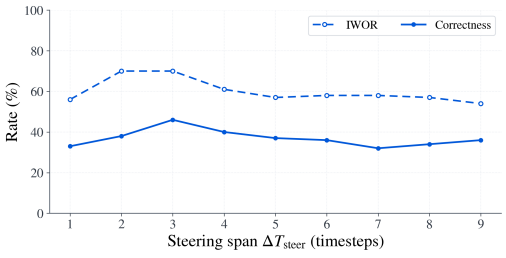

Figure 19: Correctness and IWOR across steering spans ΔT_{steer} on PersonaPlex, with the steering layer fixed to 23 and $\alpha = 5.5$. At $\Delta T_{\text{steer}} = 3$, both metrics achieve the best performance.

This metric s_t quantifies how strongly later hidden states attend back to the user input at timestep t. We use it to examine whether injecting the perception vector $\mu_{g \to p}$ restores attention to the beginning of the interrupting utterance.

We compute s_t on ZBB examples under three conditions: *no-interruption*, *interruption*, and *interruption with steering*. The heatmaps are aligned to the beginning of the zero-buffer query, allowing us to compare how much attention the model allocates to the earliest timesteps of the interruption.

Figure 20 shows that s_t decreases in the *interruption* condition, especially near the beginning of the zero-buffer query. After injecting the perception vector, s_t in the *interruption with steering* condition increases substantially relative to the *interruption* condition and approaches the level of the *no-interruption* condition. This result suggests that the perception vector helps restore attention to the earliest timesteps of the interrupting user input, providing additional evidence that steering mitigates state inertia at the attention level.

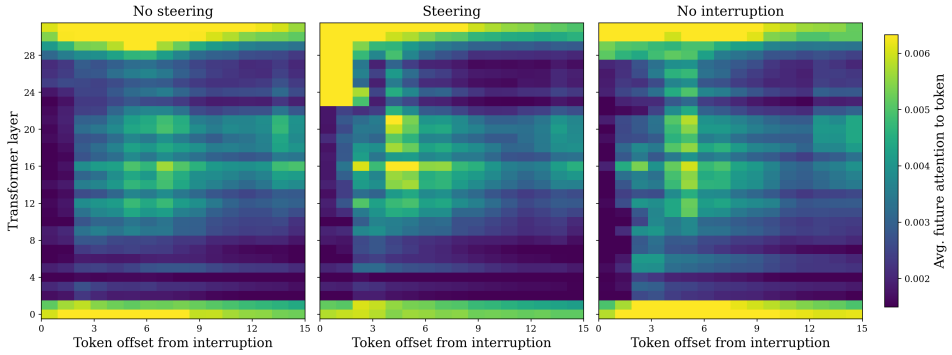

Figure 20: Attention recovery after steering. Heatmaps show the average attention weight assigned to each interruption timestep t by subsequent timesteps at varying offsets. Attention around the 5th timestep corresponds to the first semantic word of the zero-buffer query. **Left:** In the *interruption* condition, attention to the beginning of the zero-buffer query is reduced, consistent with degraded correctness and IWOR. **Middle:** In the *interruption with steering* condition, injecting the perception vector $\mu_{g \to p}$ restores attention to the earliest interruption timesteps. **Right:** In the *no-interruption* condition, the model allocates strong attention to the beginning of the zero-buffer query.

H Full-Duplex Bench Results

We also evaluate activation steering on Full-Duplex Bench (FDB) [24] to test its effect on broader full-duplex dialogue performance. We use the FDB user-interruption evaluation, which scores model responses to interruption queries on a 1–5 scale using GPT-4-Turbo. As shown in Table 5, steering preserves the score within uncertainty, suggesting that the perception vector does not degrade general full-duplex response quality.

One reason is that FDB interruption queries often contain a leading filler or attention-getting phrase before the core semantic content. For example, queries such as "Let's switch to talking about laptops" or "Hold on, what time is the meeting scheduled today?" provide several initial words before the main content needed to answer the query. Therefore, unlike ZBB, FDB does not require the model to process the core semantic content immediately after interruption. By the time the core content appears, the model may have already transitioned toward the perceptive state, making FDB less sensitive to state inertia.

I Robustness to False Triggers

We evaluate the robustness of activation steering to false trigger events. Since steering is applied at the detected interruption onset, an incorrect trigger could inject the perception vector when no real interruption occurs. To simulate this failure mode, we randomly inject the perception vector at incorrect timesteps while the model answers ZBB queries, and evaluate the resulting response quality using GPT-4.1-mini on a 1–5 scale.

Table 5: Full-Duplex Bench results before and after steering, using our reproduction of the original FDB setup.

Model	Method	FDB
PersonaPlex	Interrupt	3.34 ± 0.08
	Interrupt+Steer	3.41 ± 0.08
Moshi	Interrupt	3.45 ± 0.08
	Interrupt+Steer	3.36 ± 0.08
Raon	Interrupt	2.41 ± 0.09
	Interrupt+Steer	2.41 ± 0.09

As shown in Figure 21, response quality degrades gradually as false triggers become more frequent. This suggests that the method is tolerant to occasional false triggers, but accurate interruption detection remains important for deployment. Semantic-aware interruption detection or VAD systems can reduce this risk by distinguishing semantically meaningful speech from non-semantic acoustic events [43, 3].

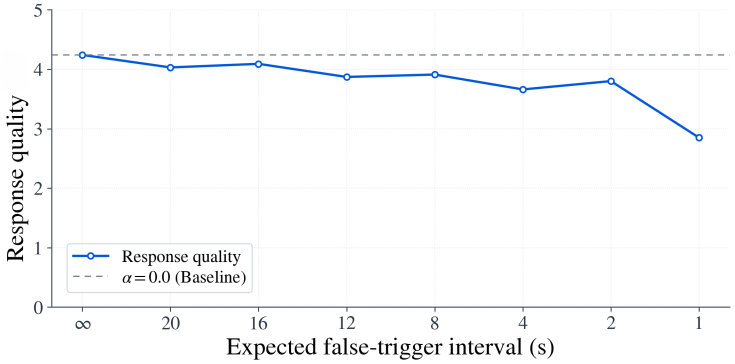

Figure 21: Response quality under false steering triggers. The x-axis represents the expected interval between false triggers. Response quality gradually decreases as false triggers become more frequent.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction accurately reflect the paper's contributions and scope. The claims are supported by the proposed diagnostic analysis, ZBB evaluation, and activation steering results.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss limitations in Section 7, including interruption onset detection, false triggers, limited availability of open-source FD-SLMs, and the diagnostic nature of logit-lens-based affinity scores.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: This paper does not present theoretical results. The quantities defined in the paper are used for diagnostic analysis and empirical evaluation.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide the information needed to reproduce the main experiments, including dataset construction, the LLM and TTS models used for dataset generation, the ASR and LLM models used for evaluation, the evaluation prompts, steering hyperparameters, and the FD-SLMs evaluated. All evaluated FD-SLMs are open-source models.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide code to reproduce the experiments in the paper, together with the Zero-Buffer Benchmark data. The released repository includes instructions for running the evaluation and reproducing the main results.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The proposed method is training-free and does not require optimization or fine-tuning. We provide the test-time setup, dataset generation procedure, evaluation protocol, LLM-judge prompts, and steering hyperparameters in the main text and appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report uncertainty values for the main experimental results. The table captions specify that these values correspond to one standard error, computed from the Bernoulli standard error $\sqrt{p(1-p)/n}$ over 100 ZBB examples.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We describe the computational resources used for the experiments in Appendix B, including the GPU type and the fact that the experiments require inference only.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms to the NeurIPS Code of Ethics. The experiments use open-source models and synthetically generated evaluation data, and do not involve human subjects or private user data.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [N/A]

Justification: This paper is primarily a diagnostic and mechanistic study of existing FD-SLMs, focusing on hidden-representation analysis and interruption handling rather than deployment. It does not introduce a new user-facing system, train or release a new model, or use private user data; therefore, we do not include a separate broader-impact discussion.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: This paper does not release new high-risk models, scraped datasets, or user-facing generative systems. The work studies existing open-source FD-SLMs and proposes an inference-time analysis and steering method for interruption handling.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We cite the open-source FD-SLMs, TTS/ASR models, and codebases used in our experiments, and document their URLs and licenses in the released repository. The codebase includes third-party components under the MIT License and Apache License 2.0, and their original notices are preserved. Our released code includes a top-level LICENSE file and third-party license notices.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We have README.md for our code and dataset bundle.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: This paper does not include any crowdsourcing experiment.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing experiments or research involving human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: We describe the use of LLMs for dataset generation and evaluation. In particular, we specify the LLM used to generate synthetic utterances and provide the prompts used for LLM-based evaluation.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.