# FINE-GRAINED DEVICE-DIRECTED SPEECH RECOGNITION: FROM OFFLINE MODELING TO STREAMING INFERENCE

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

A smart speaker in a shared space must select device requests from mixed conversation, adapt to changing functions, and preserve information beyond the transcript as interaction unfolds. We introduce fine-grained device-directed speech recognition (DDSR), which jointly transcribes semantic units and identifies their intents. An offline realization establishes this recognition and selection interface, with contextual verification for unresolved intents. Because selection depends on the device's supported functions and users' expressions, dialogue-based intent management updates the candidate inventory and registers difficult expressions without changing model weights. To retain information that transcription alone omits, the interface associates each semantic unit with speaker, emotion, and acoustic-event information. We then extend the interface to continuous input through fine-grained streaming device-directed speech recognition (FSDR). Intent-delimited unit completion enables incremental selection, while parallel attribute heads provide rich outputs without inserting additional attribute tokens into autoregressive transcription. On English and Chinese evaluations, offline DDSR achieves recording-level equal error rates of 1.25% and 5.46%, compared with 2.74% and 6.26% for a reproduced SELMA DDSD baseline; FSDR achieves 4.60% and 9.82%. Further evaluations assess dynamic intent management, acoustic-event classification, emotion recognition, and speaker attribution, connecting fine-grained content selection with configurable, informative, and incremental outputs for voice assistants.

## 1 INTRODUCTION

Consider a smart speaker in a shared living room. A user says, "Set the air conditioner to 26 degrees. Have you had lunch? Turn on the fan," interleaving device requests with a question to another person. Conventional device-directed speech detection (DDSD) assigns one acceptance decision to an input audio segment (Mallidi et al., 2018; Wagner et al., 2025). Accepting the entire input retains the side conversation; rejecting it loses both commands. The assistant needs a finer-grained interface that identifies and transcribes the individual units it should process (Figure 1).

We establish this interface through an offline realization of *fine-grained device-directed speech recognition* (DDSR). An audio–language model jointly transcribes semantic units and assigns intents, using linguistic understanding to connect each unit to a supported function. Intent predictions guide content selection rather than imposing one decision on the recording. Some requests, such as "a little lower," remain ambiguous without preceding dialogue. A contextual verifier therefore revisits its units whose intents are unresolved, using history to recover requests that first-stage recognition misses. This gives the assistant a structured account of what was said and which units should be processed.

The selection criterion, however, depends on what the device currently supports. A request for a newly added function may fall outside a fixed intent inventory, while an unfamiliar expression may obscure an existing function. We address these two sources of change through *dynamic intent management*: a persistent management intent enables dialogue-based addition and deletion of candidate



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

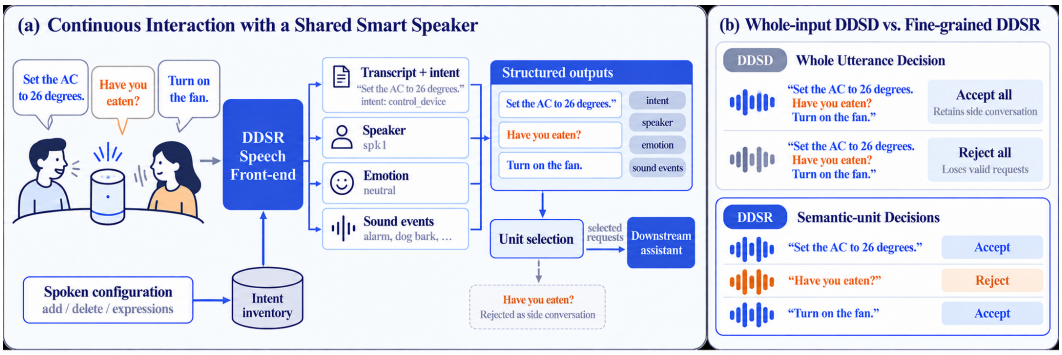



Figure 1: Motivating smart-speaker interaction. Device requests and side conversation share an audio input. DDSR supplies semantic-unit decisions and structured attributes for a downstream assistant; spoken configuration updates the recognition inventory. Whole-input acceptance cannot separately retain the requests and reject the intervening conversation.

intents, and an expression registry associates difficult formulations with their intended functions. Both recognition and contextual verification use the updated inventory; expression matching uses the updated registry, without changing model weights. Thus, the selection interface can adapt as device capabilities and user expressions evolve.

Selecting the right content still leaves an information gap. A transcript does not explicitly indicate which speaker produced a request, the emotion conveyed by the voice, or an accompanying alarm. These cues can inform how a downstream assistant interprets and responds to the selected content. We therefore extend the structured interface with speaker, emotion, and acoustic-event information associated with each semantic unit. The offline model generates these fields within a unified response, preserving information beyond words alongside fine-grained recognition.

Finally, ongoing interaction requires the same interface to deliver completed units while audio continues to arrive. Waiting for an entire recording would defer both selection and access to its associated attributes. Our *fine-grained streaming device-directed speech recognition* framework (FSDR) extends DDSR to causal processing with intent-delimited unit completion. An opening intent delimiter triggers parallel speaker, emotion, and acoustic-event predictions; the closing delimiter completes the transcript–intent pair for acceptance or contextual verification. This design provides incremental unit decisions and rich outputs without inserting autoregressive attribute tokens before the next unit's transcription.

Together, these components address what to select, how selection adapts, what information to retain, and when to deliver it. Our contributions are:

- A fine-grained DDSR interface for transcription and semantic-unit selection, established offline through intent recognition and contextual verification.
- Dialogue-based intent management and rich outputs that adapt the interface to changing functions and expressions while retaining speaker, emotion, and event information.
- A streaming realization with intent-delimited unit completion and parallel attributes, supported by data construction and evaluations of English/Chinese recognition, runtime customization, and rich outputs.

## 2 Related Work

**Device-directed speech detection.** DDSD integrates acoustic embeddings, ASR hypotheses, and decoder features to distinguish device requests from background speech (Mallidi et al., 2018). Language-model approaches combine acoustic and textual evidence (Wagner et al., 2024); SELMA jointly supports DDSD, ASR, and other assistant-input tasks (Wagner et al., 2025). DDSR changes the output from one input-level decision to transcribed semantic units with separate intents and decisions.

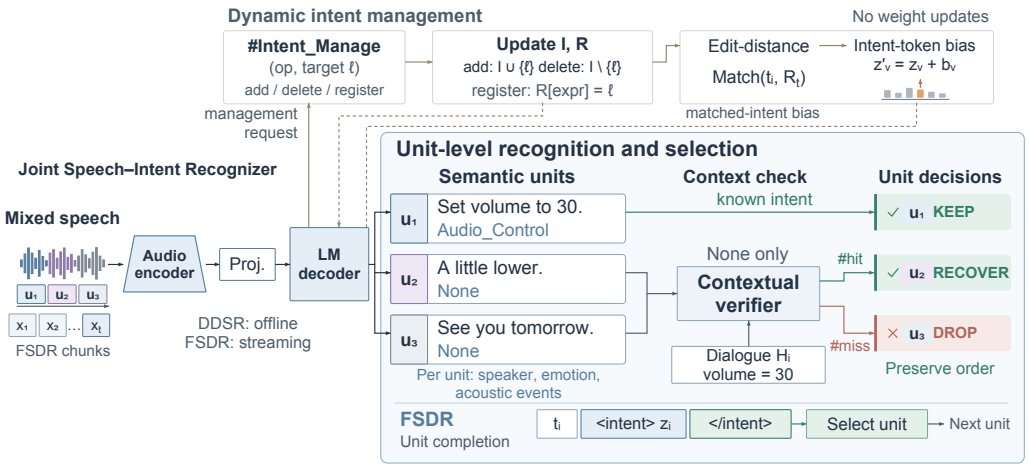



Figure 2: DDSR/FSDR architecture. Semantic units carry transcripts, intents, and rich information. Only unresolved intents undergo contextual verification. Runtime management updates intents and expression-based biases; FSDR selects units when their intent blocks close.

**Streaming recognition.** Causal DDSD updates directedness as audio arrives (Tong et al., 2021), and streaming spoken language understanding supports multiple intent predictions (Potdar et al., 2021). Voxtral Realtime provides causal audio encoding and delay-conditioned audio–text alignment (Mistral-AI et al., 2026). We adopt these backbone principles and introduce intent-delimited unit completion and attribute triggers for DDSR.

**Audio–language models and rich outputs.** Qwen-Audio combines multiple audio tasks, including emotion recognition, through language generation (Chu et al., 2023). Qwen2-Audio provides an encoder–projector–decoder architecture for instruction following (Chu et al., 2024). Whisper-AT also extracts event information from speech representations (Gong et al., 2023). We associate rich outputs with DDSR through unified generation offline and parallel heads in streaming, using MELD annotations for emotion and speaker supervision (Poria et al., 2019).

# 3 Method

## 3.1 Overview and Task Formulation

Given audio $X$, preceding dialogue $\mathcal{H}$, and an intent inventory $\mathcal{I}$, DDSR predicts an ordered sequence

$$\hat{\mathcal{U}} = \{(\hat{t}_i, \hat{z}_i, \hat{y}_i)\}_{i=1}^{N}, \tag{1}$$

where $\hat{t}_i$ is a semantic unit's transcript, $\hat{z}_i$ its first-stage intent, and $\hat{y}_i \in \{0, 1\}$ its rejection/acceptance decision. A unit can be a command, sentence, or contextual continuation; its boundary is part of recognition. Rich outputs associate each semantic unit with a speaker label $\hat{s}_i$, an emotion label $\hat{e}_i$, and acoustic-event information $\hat{c}_i$. These attributes remain linked to the same unit through recognition and selection.

Figure 2 connects recognition, contextual verification, runtime management, and rich outputs. Offline recognition consumes complete recordings; streaming recognition uses causal audio and intent-delimited units.

## 3.2 Audio–Language Recognition and Contextual Verification

**Audio–language backbone.** An audio encoder $E_a$ extracts acoustic features, and a projector $P$ maps them into the language decoder's embedding space:

$$H^a = E_a(X), \qquad H^v = P(H^a). \tag{2}$$

Following the encoder–projector–decoder organization of Qwen2-Audio (Chu et al., 2024), task instructions and candidate labels are tokenized and embedded alongside the projected audio. In the offline input template, $H^v$ replaces the audio-placeholder embeddings. Let $\rho_1(\mathcal{I}^+)$ denote the recognition prompt, where $\mathcal{I}^+ = \mathcal{I} \cup \{\texttt{Intent_Manage}\}$. The decoder models the transcript-and-intent sequence $Y^{(1)}$ as

$$p_\theta(Y^{(1)} \mid X, \rho_1) = \prod_m p_\theta(y_m^{(1)} \mid y_{<m}^{(1)}, H^v, \rho_1). \tag{3}$$

Transcription supplies the recognized content; intent generation assigns a supported function or `None`. Both use shared acoustic and linguistic representations. Backbone configurations appear in Section 4.1.

Recognition uses an audio prompt with candidate intents; verification uses the unresolved transcript, preceding dialogue, and the same current intent inventory. ASR and rich-output prompts specify their corresponding targets (Table 8 in the appendix).

**Two-stage recognition.** A recognized intent leads to acceptance. When $\hat{z}_i = \texttt{None}$, the verifier $g_\phi$ reads $\rho_2(\hat{t}_i, \mathcal{H}_i, \mathcal{I}_t^+)$ through the language pathway, without another audio-encoding pass:

$$\hat{y}_i = \begin{cases} 1, & \hat{z}_i \in \mathcal{I}^+, \\ \mathbf{1}\{g_\phi(\hat{t}_i, \mathcal{H}_i, \mathcal{I}_t^+) = \texttt{\#hit}\}, & \hat{z}_i = \texttt{None}. \end{cases} \tag{4}$$

Both stages use the current inventory $\mathcal{I}_t^+ = \mathcal{I}_t \cup \{\texttt{Intent_Manage}\}$; contextual recovery is restricted to this supported scope. Both realizations use this policy, and accepted transcripts retain their original order. The `None` label includes non-device-directed speech and requests outside the current candidate inventory; its interpretation is therefore conditioned on that inventory.

### 3.3 Streaming Recognition and Unit Completion

**Causal processing.** FSDR adapts the causal encoding and audio–text alignment principles of Voxtral Realtime (Mistral-AI et al., 2026). The encoder restricts access to current and preceding audio; projected audio representations condition frame-aligned autoregressive decoding. Delay conditioning controls the target audio–text offset. We retain these backbone mechanisms and focus on the output organization required for DDSR; numerical settings appear in Section 4.1 and Appendix A.

**Intent-delimited units.** The decoder emits

$$Y = \hat{t}_1 \texttt{<intent>}\hat{z}_1\texttt{</intent>}\ \hat{t}_2 \texttt{<intent>}\hat{z}_2\texttt{</intent>}\cdots. \tag{5}$$

Transcript tokens become available incrementally. The opening `<intent>` delimiter transitions from transcribing a unit to predicting its intent. At this transition, speaker, emotion, and acoustic-event heads read the decoder state and attach attributes to that unit (Section 3.5). These attributes do not occupy additional positions in the autoregressive output sequence.

**Unit completion.** The closing `</intent>` delimiter declares that the current transcript–intent pair is complete. A recognized intent makes the unit eligible for acceptance; a `None` intent invokes contextual verification under Equation 4. The completed unit is associated with its previously predicted speaker, emotion, and acoustic-event attributes, and decoding proceeds to the next unit. Thus, attribute prediction and acceptance occur at distinct transitions: intent onset provides the attributes, whereas intent completion supplies the label needed for the decision. No separate sentence-end classifier is introduced. For example:

```
Turn the volume down.<intent>Audio_Control</intent>
I will call you tomorrow.<intent>#None</intent>
```

Both units receive speaker, emotion, and acoustic-event predictions, although their acceptance decisions differ. Contextual verification is applied only when needed, and the resulting decision remains associated with its original unit index.

Timestamp-aligned training targets and attribute-trigger supervision are detailed in Appendix A.

### 3.4 DIALOGUE-BASED DYNAMIC INTENT MANAGEMENT

To accommodate changing functions and expressions, we maintain an editable inventory $\mathcal{I}_t$ and persistent `Intent_Manage` label in the prompt. Management requests activate this label and specify an operation and target intent.

**Inventory updates.** The model serializes each management record as

```
#Intent_Manage[op=add, target=battery_query]
```

For operation $o \in \{\text{add}, \text{delete}\}$ and target $\ell$, the runtime manager updates:

$$\mathcal{I}_{t+1} = \begin{cases} \mathcal{I}_t \cup \{\ell\}, & o = \text{add}, \\ \mathcal{I}_t \setminus \{\ell\}, & o = \text{delete}. \end{cases} \tag{6}$$

Subsequent recognition and verification prompts read the updated inventory, with `Intent_Manage` retained. Updates do not change model weights.

**Long-tail expression registration.** A registry $\mathcal{R}_t$ binds expressions to intents. Edit-distance matching compares a transcript with registered expressions; a match biases tokens continuing its target intent:

$$\tilde{z}_{m,v} = z_{m,v} + b_{m,v}, \tag{7}$$

where $b_{m,v} > 0$ on the matched intent-token path and is zero elsewhere. Registration changes intent selection while preserving the transcript. Both realizations use this interface for new intents and difficult expressions.

### 3.5 RICH SPEECH AND AUDIO OUTPUTS

Speaker, emotion, and acoustic-event information are all associated with semantic units. Event information describes sounds accompanying the corresponding unit. Speaker IDs follow first appearance within an input and are reused when a speaker returns.

**Offline: unified generation.** The offline audio–language decoder generates attributes as part of its structured response. The prompt includes the requested fields, emotion labels, and event inventory. For each semantic unit, speaker and emotion fields accompany the transcript, and acoustic-event information is emitted in a `<descriptions>` block after that unit's intent output. These fields receive token-level supervision through the shared decoder, without separate attribute classifiers.

**Streaming: intent-onset attribute heads.** When the current sentence's transcription ends and the decoder begins predicting `<intent>`, three parallel heads predict its speaker, emotion, and acoustic-event labels. Let $h_i^{\text{trig}}$ be the final-layer state used to predict this opening delimiter for unit $i$. The heads compute

$$p_i^q = \text{softmax}(f_q(h_i^{\text{trig}})), \qquad q \in \{\text{spk}, \text{emo}, \text{event}\}. \tag{8}$$

All three outputs are attached to unit $i$, including `None` units, without additional autoregressive tokens. Attribute prediction begins at intent onset; acceptance or contextual verification follows completion of the intent block.

### 3.6 DATA CONSTRUCTION

**Schema-guided generation.** We construct recognition and contextual-verification examples by separating a task schema from its reference content. An intent schema specifies a function, such as playback control, while sampled content supplies entities and alternative expressions. A contextual schema specifies a relation, such as correcting an earlier request, and conditions generation on a preceding dialogue. Length constraints diversify utterance forms. Generated texts are deduplicated using text embeddings; recognition texts are converted to speech with F5-TTS (Chen et al., 2025). Contextual examples retain their text and history. This procedure controls coverage of intended functions and context relations.

**Transforming DDSD data.** Existing internal DDSD examples contribute naturally occurring interactions. A subset is annotated and manually reviewed to build a reference pool $\mathcal{D}_{\text{base}}$. For an annotation query $q$, retrieval selects related demonstrations, which condition the annotation model:

$$C_q = f_{\text{ret}}(q, \mathcal{D}_{\text{base}}; \gamma, K), \qquad \hat{Y}_q = f_{\text{ann}}(q, C_q). \tag{9}$$

Here $K$ and $\gamma$ control the number and similarity of retrieved examples. The reference pool anchors intent assignments across similar expressions. Re-annotation supplies finer-grained recognition targets and contextual relations, complementing the coverage obtained by generation.

**Task organization and inventory variation.** Speech examples supervise transcription and intent assignment without dialogue history; text examples supervise contextual verification. Auxiliary ASR supplies transcription-only targets. Intent dropout removes selected categories from a training prompt and changes the corresponding unit labels to `None`, teaching recognition to depend on the current inventory. Evaluation recordings include positive-only, mixed positive/`None`, and all-`None` inputs (Section 4.1).

**Rich-output supervision.** For event supervision, ESC-50 sounds (Piczak, 2015) are mixed with synthesized speech as overlapping background audio. Event labels are associated with the corresponding semantic units; speech supplies the transcript–intent sequence that triggers the streaming attribute heads. MELD supplies utterance-level emotion labels and speaker annotations (Poria et al., 2019). We use its speech for emotion training and construct multi-speaker training inputs by concatenating MELD utterances from two or three distinct speakers, including later utterances from a speaker who has already appeared. Each constituent utterance is supervised with its corresponding input-local speaker ID, assigned in order of first appearance and reused for subsequent utterances from the same speaker. Each utterance retains its emotion label and is assigned `None` for DDSR training under our non-device-directed dialogue policy. Examples are mixed with recognition data; an unannotated attribute is excluded from its task loss rather than assigned a negative label. Optimization and training configurations are reported with the experiments.

# 4 Experiments

## 4.1 Experimental Setup

**Datasets.** Table 1 summarizes the intent, context, and auxiliary ASR collections. Auxiliary ASR uses 283,971 prepared English LibriSpeech entries and 800,000 prepared Chinese AISHELL-1 entries before validation splitting. We evaluate DDSR and the DDSD baselines on English and Chinese test sets containing three types of recordings: (1) recordings containing only positive reference units; (2) recordings mixing positive reference units with `None` units; and (3) recordings containing only `None` units. A reference unit is positive when its intent label is non-`None`. The `None` label covers both invalid or non-device-directed utterances and requests whose intents are absent from the prompt's candidate inventory. At the recording level, the first two types are labeled positive, while the third type is labeled negative. Management evaluation covers new intents, operation parsing, and expression recovery in both configurations. Acoustic-event inputs combine synthesized speech with overlapping ESC-50 background sounds (Piczak, 2015), preserving its five predefined folds. Speaker attribution uses multi-speaker inputs assembled from MELD utterances and is measured by speaker-label accuracy. Emotion recognition uses the speech modality of MELD (Poria et al., 2019) with seven emotion categories; its reported metric is utterance-level accuracy.

**Models and baselines.** **SELMA (DDSD, reproduced)** is our implementation of SELMA's DDSD task (Wagner et al., 2025). It uses the Qwen-Audio encoder with a 0.5B language decoder, matching DDSR in decoder parameter scale; the remaining reproduction follows the SELMA design. It uses the same speech training data as DDSR, converted to transcript-plus-binary targets, excluding text-only contextual-verification examples. Its transcript-plus-`yes`/`no` output provides a speech–language comparison with intent-based DDSR. **Offline DDSR** combines a pretrained Whisper encoder (Radford et al., 2023), a modality projector, and a Qwen2.5-0.5B decoder (Qwen et al., 2024). **FSDR** uses the Voxtral-based streaming model. **Streaming DDSD** uses the same speech–language network as FSDR but is trained to output transcription and a binary DDSD decision. This task-level baseline complements the offline SELMA comparison and the offline DDSR reference.

Table 1: Training, validation, and test collection sizes. Generated denotes LLM-generated data, synthesized as audio when used in the speech modality; re-annotated denotes DDSD examples relabeled for DDSR.

| Collection | Source | Training | Validation | Test |
|---|---|---|---|---|
| Intent | Generated | 572,597 | 5,784 | 59,668 |
| | Re-annotated | 2,023,283 | 20,437 | 19,249 |
| Context | Generated | 591,888 | 5,979 | – |
| | Re-annotated | 1,034,218 | 10,447 | 40,000 |
| Auxiliary ASR | Public | 1,073,131 | 10,840 | 35,467 |

Table 2: Offline DDSR and the reproduced SELMA DDSD baseline.

| Language | Model | EER(%, ↓) | CER(%, ↓) | WER(%, ↓) |
|---|---|---|---|---|
| English | SELMA (DDSD, reproduced) | 2.74 | 7.12 | 7.87 |
| | Offline DDSR | **1.25** | **4.79** | **5.56** |
| Chinese | SELMA (DDSD, reproduced) | 6.26 | 31.58 | 31.21 |
| | Offline DDSR | **5.46** | **28.81** | **28.72** |

**Implementation and objectives.** Both realizations use a Qwen2.5-0.5B decoder to limit language-model storage and computation. FSDR projects four encoder frames into one 80-ms position, with a six-step target alignment offset. Offline training uses MS-Swift (Zhao et al., 2024), LoRA (Hu et al., 2021), and AdamW (Loshchilov & Hutter, 2017). Recognition, verification, ASR, and generated attributes use autoregressive cross-entropy. Streaming uses weighted frame-aligned cross-entropy and z-loss, with class-label losses for annotated attributes. Speaker, emotion, and event losses are evaluated at intent onset. Missing attribute labels are masked. Training stages, weights, and hardware are detailed in Appendix A.

**Metrics.** Recording-level EER compares DDSD and DDSR using identical binary reference labels. DDSD supplies a binary acceptance score. DDSR averages confidence-weighted sentence votes, replacing first-stage votes with verifier votes only for `None` predictions, then sweeps the acceptance threshold (Appendix C). CER and WER count transcript substitutions, deletions, and insertions relative to reference length, excluding structured fields. Lower EER/CER/WER is better. Management metrics are held-out-intent EER, joint operation/target parsing accuracy, and expression-recovery rate. Events, emotion, and speaker attribution use accuracy (higher is better).

### 4.2 Offline DDSR versus DDSD

Table 2 compares offline DDSR with reproduced SELMA. EER decreases from 2.74% to 1.25% on English and from 6.26% to 5.46% on Chinese: reductions of 1.49 and 0.80 percentage points. Both CER and WER also decrease, showing lower detection and transcription errors for DDSR in this comparison.

### 4.3 Streaming DDSR

Compared with same-backbone streaming DDSD (Table 3), FSDR lowers EER by 0.46 percentage points on English and 2.02 on Chinese, while increasing CER and WER in both languages. This indicates a trade-off between directedness detection and transcription accuracy. Relative to offline DDSR, FSDR has higher EER in both languages, higher English transcription errors, and lower Chinese transcription errors; the offline reference uses a different backbone and training procedure.

### 4.4 Ablation Studies

Table 4 ablates offline components. For contextual-verification removal, we retrain without its objective and disable the verifier at inference. This raises English EER but lowers Chinese EER. Removing auxiliary ASR substantially increases WER in both languages; on Chinese, EER nev-

Table 3: Offline reference and streaming DDSD/DDSR results (%).

| Language | Configuration | EER↓ | CER↓ | WER↓ |
|---|---|---|---|---|
| English | Offline DDSR | 1.25 | 4.79 | 5.56 |
| | Streaming DDSD (same backbone) | 5.06 | 8.62 | 8.98 |
| | FSDR | 4.60 | 9.83 | 9.24 |
| Chinese | Offline DDSR | 5.46 | 28.81 | 28.72 |
| | Streaming DDSD (same backbone) | 11.84 | 18.85 | 19.62 |
| | FSDR | 9.82 | 21.85 | 22.53 |

Table 4: Offline DDSR ablations (%).

| Language | Configuration | EER↓ | CER↓ | WER↓ |
|---|---|---|---|---|
| English | Full, 0.5B decoder | 1.25 | 4.79 | 5.56 |
| | w/o contextual verification | 1.57 | 5.38 | 6.18 |
| | w/o auxiliary ASR | 2.11 | 14.01 | 16.28 |
| | Full, 2B decoder | 0.95 | 4.59 | 5.20 |
| Chinese | Full, 0.5B decoder | 5.46 | 28.81 | 28.72 |
| | w/o contextual verification | 4.76 | 25.94 | 25.78 |
| | w/o auxiliary ASR | 3.68 | 71.29 | 70.82 |
| | Full, 2B decoder | 4.37 | 24.56 | 24.39 |

ertheless falls from 5.46% to 3.68%, revealing a detection–transcription trade-off. A 2B decoder lowers EER and transcription errors in both languages relative to the 0.5B model.

### 4.5 Dynamic Intent Management

We evaluate dynamic intent management in both offline and streaming configurations along three dimensions: new-intent recognition, management parsing, and long-tail expression recovery (Table 5).

**New-intent recognition.** A 27,000-sample set contains three training-held-out intents and `#None` examples. Supplying the new intents at inference yields offline/streaming EERs of 3.16%/5.37%, evaluating inventory expansion without weight updates.

**Management parsing.** On 30,000 management and negative examples, joint operation-and-target accuracy is 98.3% offline and 96.5% in streaming mode. A correct output identifies both the update and its target intent.

**Long-tail expressions.** After expression registration, offline recognition recovers 196/198 prior misses (98.99%) and streaming recognition recovers 260/263 (98.86%). The failure collections are specific to each configuration; this evaluates targeted repair of known misses.

### 4.6 Rich-output Evaluation

**Acoustic events.** We retain ESC-50's official five-fold partition when mixing its sounds with synthesized speech. In each run, event recordings from four folds supply training mixtures, and the held-out fold supplies background sounds added to existing synthetic evaluation audio. Predictions follow the DDSR unit-output procedure. On this mixed-audio task (Table 6), offline generation achieves 92.6% accuracy and the streaming intent-onset head achieves 87.7%. Whisper-AT's published 91.7% uses original ESC-50 audio and is listed only as an external reference (Gong et al., 2023).

**Speech emotion recognition.** On MELD (Table 6), offline generation achieves 52.81% accuracy and the streaming intent-onset head achieves 51.32%. Qwen-Audio's published 55.7% is included

Table 5: Dynamic intent management.

| Evaluation | Metric | Offline | Streaming |
|---|---|---|---|
| New intents (27,000) | Device-directedness EER ↓ | 3.16% | 5.37% |
| Management (30,000) | Joint operation/target accuracy ↑ | 98.3% | 96.5% |
| Registered expressions | Recovery of prior misses ↑ | 196/198 | 260/263 |

Table 6: Rich-output accuracy (%). Event mixtures preserve ESC-50's official folds. Whisper-AT is a published reference on original ESC-50 audio and is not directly comparable to the speech mixtures. Qwen-Audio is also a published reference.

| Task | Model | Test input | ACC ↑ |
|---|---|---|---|
| Events | Whisper-AT (reported) | Original ESC-50 | 91.7 |
| | Offline DDSR | Synthetic speech + ESC-50 | 92.6 |
| | FSDR | Synthetic speech + ESC-50 | 87.7 |
| Emotion | Qwen-Audio (reported) | MELD | 55.70 |
| | Offline DDSR | MELD | 52.81 |
| | FSDR | MELD | 51.32 |
| Speaker | Offline DDSR | MELD composites | 91.3 |
| | FSDR | MELD composites | 87.7 |

as an external reference (Chu et al., 2023). The streaming head produces emotion labels without additional autoregressive tokens.

**Speaker attribution.** On MELD composites containing two or three speakers with recurring speakers, offline generation achieves 91.3% speaker-label accuracy and the streaming intent-onset head achieves 87.7% (Table 6), a difference of 3.6 percentage points. Labels identify speakers locally within each input.

**Joint-training effect.** Adding all three rich-output tasks—acoustic events, emotion, and speaker attribution—increases device-directedness EER, averaged across English and Chinese, by approximately 0.2 percentage points offline and 0.1 in streaming mode relative to the corresponding model without these tasks.

## 5 Conclusion

We introduce DDSR for semantic-unit recognition and selection in mixed speech. Dynamic intent management adapts the interface to changing functions and expressions, while rich outputs retain unit-level speaker, emotion, and acoustic-event information. FSDR supports incremental selection through intent-delimited completion and parallel attributes. Experiments evaluate recognition, runtime customization, and rich outputs across the two realizations.

## Limitations

The dynamic-intent evaluation covers three held-out intents, and expression registration targets known recognition failures. Rich-output evaluations use ESC-50 and MELD, whose sound categories and conversational conditions differ from a shared living room. The smart-speaker scenario motivates the framework; broader conversational and in-device evaluations would establish its behavior in that setting. Speaker attribution is evaluated on assembled MELD inputs with local IDs.

## AI USE STATEMENT

AI tools assisted with literature discovery and language editing. Large language models were also used for synthetic text generation and annotation assistance in dataset construction, followed by text-to-speech synthesis. All reported experimental values were provided by the authors.

## REFERENCES

Yushen Chen, Zhikang Niu, Ziyang Ma, Keqi Deng, Chunhui Wang, Jian Zhao, Kai Yu, and Xie Chen. F5-TTS: A fairytaler that fakes fluent and faithful speech with flow matching. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6255–6271, Vienna, Austria, July 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-long.313. URL https://aclanthology.org/2025.acl-long.313/.

Yunfei Chu, Jin Xu, Xiaohuan Zhou, Qian Yang, Shiliang Zhang, Zhijie Yan, Chang Zhou, and Jingren Zhou. Qwen-audio: Advancing universal audio understanding via unified large-scale audio-language models, 2023. URL https://arxiv.org/abs/2311.07919.

Yunfei Chu, Jin Xu, Qian Yang, Haojie Wei, Xipin Wei, Zhifang Guo, et al. Qwen2-audio technical report, 2024. URL https://arxiv.org/abs/2407.10759.

Yuan Gong, Sameer Khurana, Leonid Karlinsky, and James Glass. Whisper-at: Noise-robust automatic speech recognizers are also strong general audio event taggers. In *INTERSPEECH 2023*, pp. 2798–2802. ISCA, August 2023. doi: 10.21437/interspeech.2023-2193. URL http://dx.doi.org/10.21437/Interspeech.2023-2193.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2017. URL https://arxiv.org/abs/1711.05101.

Sri Harish Mallidi, Roland Maas, Kyle Goehner, Ariya Rastrow, Spyros Matsoukas, and Björn Hoffmeister. Device-directed utterance detection. In *Interspeech 2018*, pp. 1225–1228. ISCA, September 2018. doi: 10.21437/interspeech.2018-1531. URL http://dx.doi.org/10.21437/Interspeech.2018-1531.

Mistral-AI, Alexander H. Liu, Andy Ehrenberg, Andy Lo, Chen-Yo Sun, Guillaume Lample, et al. Voxtral realtime, 2026. URL https://arxiv.org/abs/2602.11298.

Karol J. Piczak. Esc: Dataset for environmental sound classification. In *Proceedings of the 23rd ACM international conference on Multimedia*, MM '15, pp. 1015–1018. ACM, October 2015. doi: 10.1145/2733373.2806390. URL http://dx.doi.org/10.1145/2733373.2806390.

Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. MELD: A multimodal multi-party dataset for emotion recognition in conversations. In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 527–536, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1050. URL https://aclanthology.org/P19-1050/.

Nihal Potdar, Anderson Raymundo Avila, Chao Xing, Dong Wang, Yiran Cao, and Xiao Chen. A streaming end-to-end framework for spoken language understanding. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, IJCAI-2021, pp. 3906–3914. International Joint Conferences on Artificial Intelligence Organization, August 2021. doi: 10.24963/ijcai.2021/538. URL http://dx.doi.org/10.24963/ijcai.2021/538.

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, et al. Qwen2.5 technical report, 2024. URL https://arxiv.org/abs/2412.15115.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine Mcleavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 28492–28518. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/radford23a.html.

Xiaosu Tong, Che-Wei Huang, Sri Harish Mallidi, Shaun Joseph, Sonal Pareek, Chander Chandak, Ariya Rastrow, and Roland Maas. Streaming reslstm with causal mean aggregation for device-directed utterance detection. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pp. 659–664. IEEE, January 2021. doi: 10.1109/slt48900.2021.9383607. URL http://dx.doi.org/10.1109/SLT48900.2021.9383607.

Dominik Wagner, Alexander Churchill, Siddharth Sigtia, Panayiotis Georgiou, Matt Mirsamadi, Aarshee Mishra, and Erik Marchi. A multimodal approach to device-directed speech detection with large language models. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 10451–10455. IEEE, April 2024. doi: 10.1109/icassp48485.2024.10446224. URL http://dx.doi.org/10.1109/ICASSP48485.2024.10446224.

Dominik Wagner, Alexander Churchill, Siddharth Sigtia, and Erik Marchi. Selma: A speech-enabled language model for virtual assistant interactions, 2025. URL https://arxiv.org/abs/2501.19377.

Yuze Zhao, Jintao Huang, Jinghan Hu, Xingjun Wang, Yunlin Mao, Daoze Zhang, et al. Swift:a scalable lightweight infrastructure for fine-tuning, 2024. URL https://arxiv.org/abs/2408.05517.

# A Implementation Details

Table 7: Architecture and optimization settings. Backbone parameter counts exclude the auxiliary classification heads.

| Component | Setting |
|---|---|
| Offline backbone | Whisper encoder + Qwen2.5-0.5B decoder |
| FSDR audio backbone | Voxtral Realtime causal encoder, approximately 970M parameters |
| FSDR speech–language backbone | Approximately 1.45B parameters |
| Audio input | 16 kHz, 128-dimensional log-Mel, 100 Hz |
| Causal convolutions | Kernel size 3; second convolution stride 2 |
| Encoder | 32 layers; hidden size 1280; attention window 750 frames |
| Projector | Four-frame concatenation; $5120 \to 896 \to 896$ |
| Aligned output rate | 12.5 Hz (80 ms per position) |
| Delay conditioning | Dimension 896, base $10^4$ / $896 \to 32 \to 896$ |
| Target delay / left padding | 6 / 32 aligned positions |
| FSDR alignment stage | LR $6 \times 10^{-5}$; approximately 482M trainable parameters |
| FSDR full-backbone stage | LR $2 \times 10^{-5}$; approximately 1452M trainable parameters |

The input layout is [prefix][BOS][audio-side], with audio frame zero aligned to BOS. Inputs are $[\mathrm{BOS}] + \mathrm{targets}[:-1]$; labels are the unshifted targets. Padding extends targets to the audio-frame count plus $\tau + 16$.

**Frame-aligned target construction.** The causal encoder emits 50 frames per second. Four adjacent frames are concatenated and projected to the language embedding dimension; aligned audio and text embeddings are added position-wise. For a word ending at $e_w$ seconds, the transcript anchor is

$$f_w = \lfloor e_w / 0.08 \rfloor + \tau + L, \qquad \tau = 6, \quad L = 32. \tag{10}$$

A `[W]` token marks a word boundary and `[P]` fills otherwise empty positions. The associated intent block follows the unit endpoint. Delay conditioning follows Voxtral Realtime through a time embedding and adaptive RMS normalization (Mistral-AI et al., 2026).

**Streaming targets.** Word timestamps place transcript targets on the audio–text alignment schedule. Each intent block follows its associated unit endpoint, and delimiters are learned in the same sequence as transcript and intent tokens. Speaker, emotion, and acoustic-event supervision is attached to the state that predicts the opening delimiter, matching its inference-time trigger. Causal attention prevents this state from accessing audio beyond the available stream prefix. The frame layout and alignment offsets are specified in Appendix A.

**Attribute trigger convention.** At inference, speaker, emotion, and acoustic-event heads read the hidden state whose language-model prediction is the opening `<intent>` token. Training selects the corresponding state for each reference delimiter. In an unshifted token-index convention, a target delimiter at position $k$ is predicted from the state at $k-1$; an implementation with pre-shifted labels must preserve this same predictive alignment. Attribute outputs are associated with their unit index and are not inserted into the autoregressive token stream. Missing annotations are masked out of the corresponding classification loss.

**Optimization settings.** Offline DDSR training uses MS-Swift (Zhao et al., 2024), AdamW (Loshchilov & Hutter, 2017), and bfloat16 precision on three NVIDIA A100 80GB GPUs. The LoRA rank and alpha are set to 8 and 16, respectively, learning rate is $5 \times 10^{-4}$, per-GPU batch size is 20, gradients are accumulated over three steps, and training lasts three epochs. FSDR uses learning rates of $6 \times 10^{-5}$ for alignment and $2 \times 10^{-5}$ for full-backbone training. Both DDSR configurations use a 0.5B language decoder to limit language-model storage and computation. FSDR uses the Voxtral Realtime causal encoder and a four-frame projector ($5120 \to 896 \to 896$), yielding an 80-ms alignment step. The target alignment offset is six steps (480 ms).

**Training objectives.** Offline training jointly supervises recognition, contextual verification, and auxiliary ASR through autoregressive cross-entropy, with LoRA adaptation (Hu et al., 2021). Generated rich-output fields receive the same token-level objective when annotated. FSDR uses weighted frame-aligned cross-entropy and z-loss:

$$\mathcal{L}_{\text{frame}} = \sum_{b=1}^{4} w_b \mathcal{L}_{\text{CE},b} + \lambda_z \mathcal{L}_z. \tag{11}$$

The intent-token weight is 3.0; the `[P]/[W]` group is weighted 2.0 during alignment and 1.0 during full-backbone training. Alignment training freezes the audio encoder, followed by backbone adaptation. Streaming speaker, emotion, and acoustic-event heads receive their respective class-label cross-entropy losses at intent-onset trigger states. Only examples with the relevant labels contribute to each auxiliary loss. Original recognition data are mixed with annotated rich-output examples.

**Final checkpoints and joint rich-output training.** Within each configuration, the reported recognition, dynamic-intent-management, and rich-output results are obtained from the same final model checkpoint: one final offline DDSR model and one final streaming FSDR model. These final models incorporate acoustic-event recognition, emotion recognition, and speaker attribution alongside the original recognition tasks. Relative to the corresponding configuration without these three tasks, joint rich-output training increases device-directedness EER averaged across English and Chinese by approximately 0.2 percentage points offline and 0.1 percentage points in streaming mode. These changes describe the combined effect of all three tasks, rather than acoustic-event training alone. Baseline models and ablations use their respective checkpoints.

## B Data Construction Details

Gemini 3 Flash generates texts under intent, entity, and length constraints; embedding-based deduplication precedes F5-TTS synthesis. For DDSD re-annotation, retrieval selects manually reviewed examples from $\mathcal{D}_{\text{base}}$ to guide Qwen3-30B-A3B:

$$C = f_{\text{ret}}(X, \mathcal{D}_{\text{base}}; \gamma, K), \qquad \hat{Y} = f_{\text{ann}}(X, C). \tag{12}$$

Here $K$ and $\gamma$ denote the retrieval count and threshold.

## C Detection Scoring

**Recording-level reference labels.** EER is computed at the recording level. A recording is positive if it contains at least one reference unit with a non-`None` intent; otherwise, it is negative. These reference labels are shared by the DDSD and DDSR evaluations.

**Prediction confidence.** For sentence $j$, let $\hat{z}_j$ denote its first-stage intent prediction. Let $p_{1,j}$ be the confidence of this prediction and $p_{2,j}$ the confidence of the contextual verifier's output, when the verifier is invoked. For stage $k$, confidence is computed as

$$p_{k,j} = \exp\left(\frac{1}{M_{k,j}} \sum_{m=1}^{M_{k,j}} \ell_{k,j,m}\right), \tag{13}$$

where $\ell_{k,j,m}$ is the decoding log-probability of the $m$-th token in the predicted label following the # marker, and $M_{k,j}$ is the number of label tokens.

**First-stage votes.** Each sentence contributes a confidence-weighted vote:

$$s_j^{(1)} = \begin{cases} 0.5 + 0.5p_{1,j}, & \hat{z}_j \neq \texttt{None}, \\ 0.5 - 0.5p_{1,j}, & \hat{z}_j = \texttt{None}. \end{cases} \tag{14}$$

Thus, a confident non-`None` prediction produces a vote close to one, while a confident `None` prediction produces a vote close to zero.

Table 8: Task interfaces. Prompt descriptions summarize their contents, not literal templates. $\mathcal{H}_i$ contains preceding dialogue.

| Task | Input / prompt content | Target |
|---|---|---|
| Recognition | Audio; task instructions; $\mathcal{I}^+$ | Transcript + unit intents |
| Verification | $\hat{t}_i$; $\mathcal{H}_i$; $\mathcal{I}_t^+$; contextual relevance instruction | `#hit` / `#miss` |
| Auxiliary ASR | Audio; transcription instruction | Transcript |
| Rich outputs | Audio; requested attributes and candidate categories | Offline: generated fields; streaming: head labels |

**Contextual vote replacement.** Following Equation 4, contextual verification is invoked only for sentences whose first-stage prediction is `None`. Let $\hat{c}_j$ denote the verifier's output, either `#hit` or `#miss`. The verifier replaces the corresponding first-stage vote, giving the final sentence vote

$$s_j = \begin{cases} 0.5 + 0.5p_{1,j}, & \hat{z}_j \neq \texttt{None}, \\ 0.5 + 0.5p_{2,j}, & \hat{z}_j = \texttt{None},\ \hat{c}_j = \texttt{\#hit}, \\ 0.5 - 0.5p_{2,j}, & \hat{z}_j = \texttt{None},\ \hat{c}_j = \texttt{\#miss}. \end{cases} \tag{15}$$

For the ablation without contextual verification, $s_j = s_j^{(1)}$ for every sentence.

**Recording-level aggregation.** For a recording $X$ containing $J$ predicted sentence–intent pairs, the acceptance score is

$$S(X) = \frac{1}{J} \sum_{j=1}^{J} s_j. \tag{16}$$

All predicted sentences contribute to the average. The aggregation uses predicted labels and confidences; reference labels are used only for evaluation.

**Equal error rate.** Let $\mathcal{D}_+$ and $\mathcal{D}_-$ denote the sets of positive and negative test recordings, respectively. At threshold $\theta$, a recording is accepted when $S(X) \geq \theta$ and rejected otherwise. The false-acceptance and false-rejection rates are

$$\mathrm{FAR}(\theta) = \frac{1}{|\mathcal{D}_-|} \sum_{X \in \mathcal{D}_-} \mathbf{1}\{S(X) \geq \theta\}, \tag{17}$$

$$\mathrm{FRR}(\theta) = \frac{1}{|\mathcal{D}_+|} \sum_{X \in \mathcal{D}_+} \mathbf{1}\{S(X) < \theta\}. \tag{18}$$

We sweep $\theta$ to obtain the equal-error operating point where FAR and FRR coincide, or approximately coincide for the empirical scores.

## D Task Prompt Interfaces

**Task-specific prompts.** The recognition prompt $\rho_1$ specifies transcription and intent assignment without dialogue history. Contextual verification uses a different prompt $\rho_2(\hat{t}_i, \mathcal{H}_i, \mathcal{I}_t^+)$ containing the unresolved transcript, preceding dialogue, and current intent inventory, with `#hit`/`#miss` as its output vocabulary. ASR-only examples request transcription without intent labels, and rich-output prompts additionally specify the attributes and their label inventories. Table 8 summarizes these distinct interfaces.