TOOLDF: Tool-Integrated Reasoning for Mixed-Authenticity Audio Deepfake Detection

Anonymous ACL submission

Abstract

Audio deepfake detection is commonly formulated as clip-level binary classification within a single acoustic domain. However, real-world manipulated audio often exhibits mixed authenticity, where genuine and synthetic cues coexist across temporal transitions, overlapping sources, or both. This setting requires not only detecting manipulated audio, but also localizing the components that provide evidence for the decision. We propose ToolDF, a tool-integrated reasoning framework for mixed-authenticity audio deepfake detection. ToolDF employs an audio large language model as an orchestrator trained with supervised tool-use trajectories, enabling adaptive audio-scene decomposition, selective source separation, domain-specific expert routing, and evidence-grounded aggregation into an interpretable verdict. We further introduce a mixed-authenticity ADD benchmark covering temporal transitions, acoustic overlaps, and hybrid compositions. Experimental results show that ToolDF consistently improves composite-type detection, achieving absolute macro-F1 gains of 3.72% over the strongest monolithic baseline and 14.39% over a fixed pipeline, while providing fine-grained and interpretable evidence over temporal regions and acoustic sources. Our source code and dataset are publicly available online[1].

1 Introduction

Audio deepfake detection (ADD) is typically formulated as a clip-level binary classification problem (Jung et al., 2022; Tak et al., 2022; Rosello et al., 2023; Kim et al., 2025), where an input audio clip is predicted as either real or fake. While this formulation has supported significant progress across speech, singing, music, and environmental-sound domains (Zang et al., 2024a,b; Afchar et al., 2025; Xie et al., 2025, 2026a), it becomes restrictive when manipulated media contains multiple acoustic components with different authenticity states.

To address this limitation, we introduce the task of mixed-authenticity audio deepfake detection, where genuine and manipulated cues coexist within a single audio input. While recent efforts have begun addressing localized or domain-specific manipulations (Zhang et al., 2022; Zang et al., 2024b; Zhang et al., 2026), real-world mixed-authenticity audio may contain complex manipulations across temporal transitions, acoustic sources, or both. For example, an audio clip might feature a transition from genuine speech to synthetic singing, all layered over genuine background music. A robust detection system should therefore not only determine whether the audio is manipulated, but also precisely localize and identify the specific segments or sources that provide evidence for the final decision.

Existing ADD systems are not designed to handle such mixed-authenticity inputs. Domain-specific detectors can become unreliable when an input contains out-of-domain sources, while binary classifiers may overlook localized manipulations (Li et al., 2024). Fixed pipelines that rely on predefined separation or detection steps are also suboptimal (Zang et al., 2024b; Zhang et al., 2026), as separation can help with overlapping sources but may introduce artifacts when the input is already clean. Moreover, directly using an audio large language model (ALLM) (Gu et al., 2025; Guo et al., 2026) naively as a holistic binary classifier does not leverage specialized detectors or reveal what evidence supports the prediction. These limitations motivate an adaptive framework that analyzes the input structure, selectively invokes expert tools, and integrates their outputs into an interpretable authenticity decision.

In this paper, we propose ToolDF, a tool-

[1] https://github.com/a24897618/tooldf

integrated reasoning framework that uses an ALLM as an orchestrator rather than as a direct authenticity classifier. Given an audio input, ToolDF analyzes its structure to identify relevant temporal segments and acoustic sources. It then selectively invokes domain-specific expert detectors for speech, singing, music, or environmental sounds. When overlapping sources are detected, ToolDF applies source separation prior to localized detection. Finally, it aggregates the tool responses into an interpretable authenticity decision.

To achieve this, we train ToolDF via supervised fine-tuning (SFT) on structured tool-use trajectories. Each trajectory explicitly encodes a chain of reasoning, demonstrating how the model analyzes the input structure, plans tool calls, interprets expert responses, and deduces the final verdict. Instead of relying solely on the final authenticity label, these trajectories provide rich intermediate supervision for structural analysis, tool selection, and evidence aggregation.

To evaluate the proposed framework, we construct comprehensive mixed-authenticity audio benchmarks covering both single-type and composite-type manipulation scenarios. Experimental results demonstrate that ToolDF consistently outperforms conventional ADD models and fixed-pipeline baselines, especially when detection requires intricate reasoning over temporal segments and overlapping sources. Furthermore, ToolDF delivers highly interpretable tool-execution traces, clearly revealing which segments, sources, and detector outputs contribute to the final decision.

Our contributions are summarized as follows:

- We formulate the task of mixed-authenticity audio deepfake detection, introducing a realistic scenario where genuine and manipulated cues coexist across temporal transitions and overlapping acoustic sources.

- We propose ToolDF, an ALLM-based, tool-integrated reasoning framework that acts as an intelligent orchestrator to dynamically analyze audio structure, selectively route expert tools, and seamlessly aggregate evidence via supervised tool-use trajectories.

- We construct and publicly release comprehensive mixed-authenticity benchmarks, demonstrating through extensive experiments that ToolDF consistently outperforms existing ADD baselines while providing transparent, interpretable component-level evidence.

2 Related Work

2.1 Audio Deepfake Detection

ADD has been studied extensively in the speech domain, with progress driven by benchmarks such as ASVspoof (Yamagishi et al., 2021) and architectures like AASIST (Jung et al., 2022). Recent efforts have expanded the scope of ADD from speech to diverse acoustic domains, including singing voices, music, and environmental sounds (Zang et al., 2024a,b; Afchar et al., 2025; Xie et al., 2025, 2026a). However, these methods assume that each input clip belongs to a single dominant acoustic domain and predict a single authenticity label for the entire audio. This paradigm becomes restrictive in real-world scenarios, where multiple audio types and varying authenticity states can coexist within a single input.

2.2 Multi-Source and Composite Audio ADD

To address composite audio inputs, source separation has been explored to isolate source-dependent manipulation cues, particularly for vocals mixed with background music (Zang et al., 2024b; Zhang et al., 2026). For instance, a fixed preprocessing pipeline (Zang et al., 2024b) applies source separation across all inputs, which can introduce unwanted artifacts into already clean, single-source signals. Conversely, while recent frameworks (Zhang et al., 2026) adaptively trigger separation upon detecting mixed signals, their downstream detection is strictly limited to the speech domain, failing to account for synthetic elements in singing, music, or environmental backgrounds. Unlike these approaches with inflexible pipelines or restricted detection scopes, *ToolDF* adaptively invokes source separation only when structurally required, while comprehensively evaluating authenticity across all acoustic domains.

2.3 ALLM and Tool-Integrated Reasoning

ALLM has demonstrated remarkable capabilities in understanding speech, music, and environmental sounds (Rouditchenko et al., 2025), leading to early explorations in audio deepfake detection (Gu et al., 2025; Guo et al., 2026; Xie et al., 2026b). A naive approach is to deploy an ALLM directly as a holistic binary classifier. However, such direct pre-

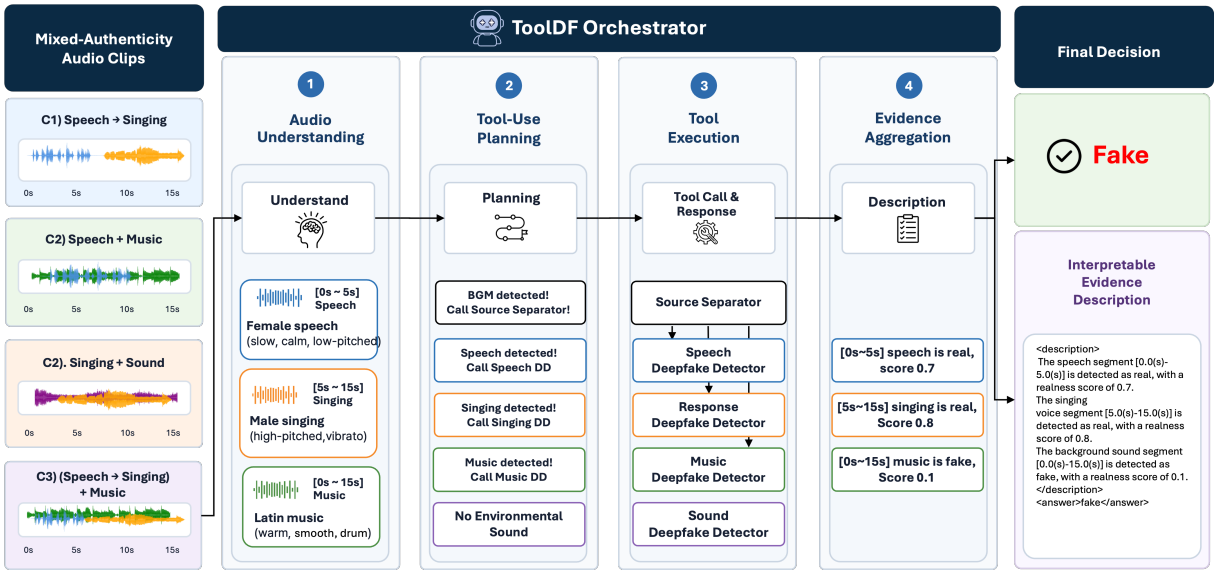

Figure 1: Overview of the ToolDF framework for mixed-authenticity audio deepfake detection.

diction operates as a black box, providing limited insight into the underlying evidence and failing to exploit specialized, domain-specific detectors. Tool-augmented language models address these limitations by invoking external modules to perform expert tasks and integrating intermediate responses into a final output (Schick et al., 2023; Qin et al., 2024; Qian et al., 2026). ToolDF leverages supervised tool-use trajectories to operationalize tool-integrated reasoning (TIR) for ADD, training an ALLM to act as an intelligent orchestrator that dynamically analyzes mixed-authenticity inputs, routes domain expert detectors, and aggregates component-level evidence through an explicit chain of reasoning.

3 Methodology

3.1 Task Formulation

Let x denote an input audio clip. Unlike conventional ADD, where x is assigned a single clip-level label, we assume that x may contain multiple heterogeneous components:

$$\mathcal{C}(x) = \{c_1, c_2, \ldots, c_K\}.$$

Each component c_k is associated with a content type $t_k \in \{\texttt{speech}, \texttt{singing}, \texttt{music}, \texttt{sound}\}$, a support region ρ_k, and a localized authenticity label $y_k \in \{\texttt{real}, \texttt{fake}\}$. When temporally localized, ρ_k is represented by onset and offset times (s_k, e_k); otherwise, it may refer to a source-specific acoustic region. The clip-level label $y \in \{\texttt{real}, \texttt{fake}\}$ follows an early-fail rule:

$$y = \texttt{fake} \iff \exists c_k \in \mathcal{C}(x) \ \text{ s.t. } \ y_k = \texttt{fake}.$$

The standard single-domain ADD problem is a restricted case where $K = 1$ and the single component spans the entire clip. In the mixed-authenticity setting, a system should predict the clip-level label y while identifying the component-level evidence that supports the decision.

3.2 ToolDF Framework

ToolDF formulates mixed-authenticity audio deepfake detection as a structured TIR process rather than direct clip-level classification. Given an input audio clip x, an ALLM serves as an orchestrator that analyzes the audio scene, plans tool usage, invokes domain-specific expert tools, and aggregates their observations into a final authenticity decision. As illustrated in Figure 1, the framework consists of four phases: audio understanding, tool-use planning, localized tool execution, and evidence aggregation.

Specifically, ToolDF implements TIR as a trajectory that links audio-scene decomposition with authenticity assessment. Rather than relying on unconstrained free-form rationales, ToolDF represents the reasoning process as an explicit sequence of structured decisions: identifying the constituent acoustic components $\mathcal{C}(x)$, assigning each component c_k to suitable expert tools based on its content type t_k, executing localized tool calls over the support region ρ_k, and deriving the final clip-level verdict y from component-level evidence. Figure 2

ToolDF Rollout

User Prompt <audio> Is this audio real or fake?

Audio Understanding and Planning
<think> <audio_understanding> [0.63(s)-2.34(s)] A female speech at a slow pace in a calm, low-pitched tone. [2.43(s)-8.37(s)] A male singing voice with a clear, high-pitched tone and gentle vibrato. [0.0(s)-8.5(s)] A continuous high-pitched electronic beep with a steady tone. Environmental sound/noise detected throughout.
</audio_understanding>
<plan>
1. Background sound overlaps with vocals. Call `source_separator` first.
2. [0.63(s)-2.34(s)] Speech detected. Call `speech_deepfake_detector` on the separated vocals.
3. [2.43(s)-8.37(s)] Singing voice detected. Call `singing_deepfake_detector` on the separated vocals.
4. [0.0(s)-8.5(s)] Background sound. Call `sound_deepfake_detector` on the separated background.
</plan> </think>

Tool Execution
<tool_call> {"name": "source_separator", "parameters": {"audio_path": "input_audio", "vocals_path": "vocals.wav", "background_path": "background.wav"}} </tool_call> <tool_response> {"model": "source_separator", "status": "completed"} </tool_response>
<tool_call> {"name": "speech_deepfake_detector", "parameters": {"start": 0.63, "end": 2.34, "vocals_path": "vocals.wav"}} </tool_call> <tool_response> {"prediction": "real", "score": 0.99} </tool_response>
<tool_call> {"name": "singing_deepfake_detector", "parameters": {"start": 2.43, "end": 8.37, "vocals_path": "vocals.wav"}} </tool_call> <tool_response> {"prediction": "fake", "score": 0.29} </tool_response>
<tool_call> {"name": "sound_deepfake_detector", "parameters": {"start": 0.0, "end": 8.5, "background_path": "background.wav"}} </tool_call> <tool_response> {"prediction": "real", "score": 0.95} </tool_response>

Evidence Aggregation
<description> The speech segment [0.63(s)-2.34(s)] is detected as real, with a realness score of 0.99. The singing voice segment [2.43(s)-8.37(s)] is detected as fake, with a realness score of 0.29. The background sound segment [0.0(s)-8.5(s)] is detected as real, with a realness score of 0.95. </description>
<answer>fake</answer>

Figure 2: Execution rollout of the structured tool-integrated reasoning framework.

shows an example rollout of this structured trajectory.

Audio Understanding. Given an input audio clip x, the orchestrator first produces a structured `<audio_understanding>` block that summarizes the acoustic composition of the clip. The block identifies a set of components $\mathcal{C}(x) = \{c_1, \ldots, c_K\}$, where each component c_k is associated with a content type $t_k \in \{\texttt{speech}, \texttt{singing}, \texttt{music}, \texttt{sound}\}$ and a support region ρ_k. This step helps characterize the acoustic structure of the input and informs the selection and use of downstream detectors.

Planning. Based on the identified components $\mathcal{C}(x)$, the orchestrator formulates a conditional tool-use sequence within the `<plan>` block. If vocal and background components overlap within the same region ρ_k, the plan prioritizes the invocation of a source separator to isolate distinct acoustic streams. The resulting individual components are then systematically mapped to the corresponding expert detectors according to their respective content types t_k.

Tool Execution. The orchestrator executes the plan by issuing structured tool calls. As shown by the `<tool_call>` and `<tool_response>` sequences in Figure 2, each call specifies which tool to use and what input region to analyze, such as a temporal segment or a separated source. The called tool then returns a localized authenticity prediction y_k for the target component. These tool outputs provide external evidence for the final decision.

Evidence Aggregation. After receiving the tool observations, the orchestrator summarizes the component-level evidence within the `<description>` block and derives the final clip-level decision. Following the mixed-authenticity formulation in Section 3.1, the final output in the `<answer>` tag is `fake` if any component-level prediction y_k indicates a synthesized or manipulated source. Otherwise, the framework returns `real`.

3.3 Supervised Trajectory Learning

We optimize ToolDF with supervised fine-tuning (SFT) to teach the ALLM the serialized format of the TIR protocol. Given an input audio clip x, the orchestrator is trained to autoregressively generate

Type	Acoustic Configurations	# Train	# Dev	# Eval
ASVspoof2019 LA (Nautsch et al., 2021)	Speech	25,380	24,844	71,237
CtrSVDD (Zang et al., 2024a)	Singing Voice	84,404	43,625	92,769
EnvSDD (Yin et al., 2025)	Environmental Sound	139,055	39,710	39,768
FakeMusicCaps (Comanducci et al., 2025)	Music (Fake)	16,055	5,350	6,200
MusicCaps (Agostinelli et al., 2023)	Music (Real)	3,211	1,070	1,071
C1	Speech → Singing	12,690	12,422	35,616
(Temporal Transition)	Singing → Speech	12,690	12,422	35,620
C2	Speech + Music	3,170	2,325	3,114
(Acoustic Overlap)	Singing + Music	10,281	4,095	4,157
	Speech + Environmental Sound	22,210	14,060	17,234
	Singing + Environmental Sound	74,122	25,650	22,534
C3	(Speech → Singing) + Music	1,418	1,197	979
(Transition + Overlap)	(Singing → Speech) + Music	1,491	1,222	985
	(Speech → Singing) + EnvSound	11,272	7,815	5,061
	(Singing → Speech) + EnvSound	11,199	7,790	5,059
Total	All Configurations	428,648	203,597	341,404

Table 1: Statistics of the mixed-authenticity audio deepfake detection benchmark, including both single-type corpora and constructed composite-type splits. The composite-type setting consists of **C1** temporal speech–singing transitions, **C2** acoustic overlaps between foreground and background sources, and **C3** hybrid mixtures combining transitions and overlaps. Details of the data construction are provided in Appendix A.

a structured trajectory from audio understanding to the final decision:

$$o = \langle u, p, \mathcal{T}, d, y \rangle,$$

where u denotes the audio-understanding block, p the tool-use plan, $\mathcal{T} = ((a_i, r_i))_{i=1}^{N}$ the sequence of tool calls a_i and tool observations r_i, d the evidence summary, and $y \in \{\texttt{real}, \texttt{fake}\}$ the final clip-level verdict.

To construct supervised trajectories, we use ground-truth component annotations:

$$\mathcal{S}^{\star} = \{(\rho_k, t_k, y_k)\}_{k=1}^{K},$$

where ρ_k, t_k, and y_k follow the definitions in Section 3.1. These annotations are used to construct audio-understanding blocks, tool-use plans, and localized tool calls. The tool observations are then inserted into the trajectory as conditioning context for subsequent generation. The evidence summary and final verdict are constructed from these observations according to the mixed-authenticity early-fail rule.

Although the trajectory contains both generated tokens and tool observation tokens, the model is optimized only on tokens produced by the orchestrator. Let $\mathcal{M}$ denote the set of token positions corresponding to u, p, $\{a_i\}_{i=1}^{N}$, d, and y, excluding the observation tokens $\{r_i\}_{i=1}^{N}$. The SFT objective is:

$$\mathcal{L}_{\text{SFT}} = -\sum_{m \in \mathcal{M}} \log P_\theta \left(o_m \mid o_{<m}, x, q \right).$$

Here, x denotes the input audio clip, q denotes the textual instruction context, o_m denotes the m-th token in the target trajectory, $o_{<m}$ denotes the preceding tokens, and P_θ denotes the next-token distribution of the ALLM.

4 Mixed-Authenticity ADD Benchmark

To evaluate ToolDF beyond isolated single-domain settings, we construct a mixed-authenticity ADD benchmark by combining public audio deepfake datasets from multiple acoustic domains. Existing benchmarks typically focus on a single content type, such as speech, singing, music, or environmental sound, and thus do not sufficiently evaluate robustness to heterogeneous audio scenes containing multiple components with different authenticity cues.

The benchmark is built from authentic and synthetic subsets of domain-specific corpora, including ASVspoof2019 (Nautsch et al., 2021) for speech, CtrSVDD (Zang et al., 2024a) for singing, FakeMusicCaps (Comanducci et al., 2025) and MusicCaps (Agostinelli et al., 2023) for music, and EnvSDD (Yin et al., 2025) for environmental sounds. Using these sources, we construct both single-type and composite-type evaluation settings. The single-type setting contains isolated speech, singing, music, and environmental-sound clips following conventional ADD evaluation. The composite-type setting consists of three splits: **C1**, temporal transitions between speech and singing; **C2**, acoustic overlap between foreground vocal

content and background music or environmental sound; and **C3**, hybrid mixtures combining both temporal transitions and acoustic overlap.

5 Experiments

5.1 Dataset

Table 1 summarizes the statistics of our mixed-authenticity ADD benchmark. We evaluate ToolDF under both conventional single-type and the proposed composite-type settings. For the single-type evaluations, we follow the official training, development, and test splits of each source corpus. As an exception, due to the archival unavailability of certain original audio files in MusicCaps (Agostinelli et al., 2023), a small number of real samples could not be retrieved during data collection; to prevent data leakage, all corresponding fake pairs from that specific subset were exclusively assigned to the test set. The remaining accessible samples from MusicCaps and FakeMusicCaps (Comanducci et al., 2025) were randomly partitioned into training, development, and test sets with a $6:2:2$ ratio. Comprehensive configurations for the composite-type settings are provided in Appendix A.

5.2 Implementation Details

ALLM Backbone. We implement ToolDF with Qwen2.5-Omni-3B (Xu et al., 2025) as the ALLM backbone and fine-tune it on supervised TIR trajectories. We use Low-Rank Adaptation (LoRA) for parameter-efficient tuning, keeping the base model frozen and inserting adapters into all linear layers with rank $r = 64$ and scaling factor $\alpha = 16$. The model is fine-tuned using the task-specific system prompt designed to enforce the structured TIR trajectory, the exact text of which is detailed in Appendix B. Training uses 428,648 examples from the union of single-type and composite-type data. We train for three epochs with AdamW, using a learning rate of 1×10^{-5}, weight decay of 0.1, bfloat16 precision, gradient checkpointing, DeepSpeed ZeRO-2, and a maximum sequence length of 4096. Training is conducted on 8 NVIDIA A40 GPUs with a batch size of 4 per GPU and gradient accumulation over 4 steps, yielding an global batch size of 128.

Tools. ToolDF uses external tools for source separation and domain-specific deepfake detection. For overlapping vocal-background mixtures, we employ Demucs v4 (Rouard et al., 2023) to obtain vocal and nonvocal streams. For authenticity assessment, we instantiate XLSR-AASIST expert detectors for speech, singing, music, and environmental sound, each trained on its corresponding single-type domain. To prevent information loss at temporal transitions, the detectors are trained on the full audio duration, with mini-batches dynamically repeated to the maximum sequence length within each batch. We also train separation-aware variants to handle separated inputs and potential separation artifacts.

At inference time, the orchestrator routes the target audio segment or separated source to the selected expert according to the generated tool-use plan. Each detector returns a binary authenticity prediction and a normalized confidence score in $[0, 1]$, which are inserted into the trajectory as tool observations. The decision threshold for each detector is determined by the equal error rate on its development set.

5.3 Baselines

To evaluate ToolDF in mixed-authenticity audio deepfake detection, we compare its performance against five baseline configurations, including four end-to-end models and a fixed pipeline. we evaluate against **XLSR-AASIST** (Tak et al., 2022), a standard end-to-end framework combining a self-supervised wav2vec 2.0 (Babu et al., 2022) backbone with a graph neural network, and **XLSR-Conformer** (Rosello et al., 2023), which leverages a Conformer backbone to capture both local and global temporal dependencies. We also compare against **WPT-XLSR-AASIST** (Xie et al., 2026a), a frequency-sensitive approach proposed for all-type audio deepfake detection, and **ALLM4ADD** (Gu et al., 2025), a recent ALLM fine-tuned specifically for this task.

Fixed Pipeline strictly applies source separation via Demucs v4 to every input, routing the isolated foreground and background streams to specialized *vocal* (speech/singing) and *non-vocal* (music/sound) XLSR-AASIST expert detectors. Both detectors are optimized as separator-aware models. The final verdict is determined by an early-fail rule, classifying the clip as fake if either detector predicts a synthetic origin.

5.4 Evaluation Metric

We use the macro-averaged F1 score as the clip-level evaluation metric for both single-type and

Model	Single-Type					Composite-Type			
	Speech	Singing	Sound	Music	S-Avg.	C1	C2	C3	C-Avg.
XLSR-AASIST (Tak et al., 2022)	89.60	84.82	**76.67**	95.22	**86.57**	**79.54**	51.95	42.01	**57.83**
XLSR-Conformer (Rosello et al., 2023)	90.17	83.34	72.56	92.83	84.73	74.29	48.82	38.12	53.74
WPT-XLSR-AASIST (Xie et al., 2026a)	90.16	**86.91**	69.26	82.45	82.20	68.41	47.87	31.46	49.25
ALLM4ADD (Gu et al., 2025)	**97.64**	61.56	75.54	**95.62**	82.56	62.62	**55.51**	**53.22**	57.12

Table 2: Zero-shot generalization performance of single-type trained models on composite-type mixed-authenticity inputs (Macro-F1 score (%)). S-Avg. and C-Avg. denote the average performance over single-type and composite-type settings, respectively.

Model	Interp.	Single-Type					Composite-Type			
		Speech	Singing	Sound	Music	S-Avg.	C1	C2	C3	C-Avg.
End-to-End Monolithic Models										
XLSR-AASIST (Tak et al., 2022)	×	88.40	**84.91**	69.37	92.42	83.78	89.79	73.16	71.57	78.17
XLSR-Conformer (Rosello et al., 2023)	×	88.09	82.55	71.80	90.80	83.31	87.28	74.46	71.61	77.78
WPT-XLSR-AASIST (Xie et al., 2026a)	×	83.55	78.81	**78.04**	87.52	81.98	85.00	73.73	69.87	76.20
ALLM4ADD (Gu et al., 2025)	×	**98.17**	60.94	75.00	**95.20**	82.33	72.77	64.92	57.43	65.04
Component-Level Frameworks										
Fixed Pipeline	✓*	63.72	55.58	52.91	66.25	59.62	55.57	**78.55**	68.39	67.50
ToolDF (Proposed)	✓	95.78	81.58	76.13	92.67	**86.54**	**91.21**	77.66	**76.81**	**81.89**
ToolDF (Oracle)	-	95.96	78.94	76.10	93.09	86.02	91.77	79.34	77.45	82.85

Table 3: Performance comparison on single-type and composite-type audio deepfake detection. Values are macro-F1 scores (%). S-Avg. and C-Avg. denote averages over single-type and composite-type settings, respectively. ✓* indicates partial or rule-based interpretability.

composite-type settings. Macro-F1 assigns equal weight to each class, making it suitable for evaluation under potential class imbalance.

6 Result

6.1 The Single-Type Generalization Gap

Table 2 examines whether models trained only on single-type data can generalize to composite mixed-authenticity inputs. While the baselines perform strongly on single-type cases, with XLSR-AASIST achieving the highest S-Avg. of 86.57, their performance drops sharply on composite inputs. The best C-Avg. falls to 57.83, again from XLSR-AASIST, and this degradation becomes larger as the acoustic scene becomes more complex from C1 to C3. ALLM4ADD is relatively more robust in overlapping cases, achieving the best scores on C2 and C3, but its overall performance still remains substantially below its single-type results.

These results show that strong performance on isolated acoustic domains does not ensure generalization to mixed-authenticity ADD. Since effectively handling composite scenarios requires more than simple domain classification, this distinct gap directly motivates ToolDF, which decomposes the acoustic scene through structured, tool-integrated reasoning.

6.2 Evaluation on Mixed-Authenticity ADD

Table 3 compares ToolDF with end-to-end monolithic models and component-level frameworks under a joint training setting. ToolDF achieves state-of-the-art performance on composite inputs, yielding the highest C-Avg. score of 81.89% and outperforming the strongest monolithic baseline, XLSR-AASIST, by 3.72%. These gains are consistent across all types, with ToolDF obtaining 91.21% on temporal transitions (C1) and 76.81% on hybrid mixtures (C3). Notably, although ALLM4ADD is robust in single-type settings, its performance drops sharply to 65.04% on composite inputs, proving that standard monolithic architectures struggle to resolve mixed synthetic evidence even with explicit training.

Furthermore, ToolDF significantly outperforms the Fixed Pipeline, achieving a C-Avg. score of 81.89% compared to the baseline's 67.50%. While the Fixed Pipeline scores slightly higher on acoustic overlaps (C2) with 78.55% due to its mandatory source separation, it fails on other splits where separation is unnecessary or destructive. ToolDF avoids this pitfall by adaptively deciding when to separate audio. Crucially, ToolDF provides high

Configuration	C1 (Temporal)			C2 (Overlap)			C3 (Temporal + Overlap)		
	PR	$F1_{parsed}$	$F1_{strict}$	PR	$F1_{parsed}$	$F1_{strict}$	PR	$F1_{parsed}$	$F1_{strict}$
ToolDF	**100.00**	**91.21**	**91.21**	99.99	**77.67**	**77.66**	99.98	**76.83**	**76.81**
w/o Audio Understanding	82.59	60.12	48.25	79.85	48.82	39.77	83.15	53.21	44.31
w/o Planning	96.71	45.47	44.08	84.34	24.57	20.87	97.52	35.18	34.33
w/o Tool Calling	**100.00**	75.75	75.75	**100.00**	66.49	66.49	**100.00**	63.62	63.62
w/o Description	69.90	85.06	48.57	23.69	69.77	18.57	88.87	74.43	63.62

Table 4: Ablation study under parsed-only and strict evaluation. PR denotes the parsing rate (%), representing the percentage of outputs that successfully conform to the designated trajectory format. $F1_{parsed}$ (%) is computed only on these successfully parsed outputs, while $F1_{strict}$ (%) is evaluated over the entire test set by treating unparsable outputs as incorrect predictions.

Type	Seg-$F1_{mi.}$	Seg-$F1_{ma.}$	Event-$F1_{mi.}$	Event-$F1_{ma.}$
Speech	97.76	98.18	93.42	93.64
Singing	95.58	97.74	95.19	97.19
Sound	99.95	99.97	99.92	99.96
Music	99.73	99.89	99.66	99.94
C1	95.63	97.34	93.34	94.24
C2	96.70	96.11	92.77	91.96
C3	95.26	93.63	86.44	87.62

Table 5: Fine-grained localization performance of ToolDF across evaluation subsets. Segment-level and event-level F1 scores (%) are reported with micro and macro averaging.

interpretability through explicit tool-execution trajectories, whereas monolithic models remain uninterpretable black boxes. Finally, ToolDF closely approaches its Oracle variant with a marginal gap of only 0.96%, showing that the orchestrator effectively finds optimal routing paths without ground-truth annotations.

6.3 Ablation and Localization Analysis

Table 4 reports the ablation results for composite-type inputs under both parsed-only and strict evaluation metrics. The full ToolDF model achieves the highest accuracy across all configurations while maintaining a nearly perfect parsing rate (PR) close to 100.00%. This demonstrates that our structured trajectory design generates stable, rule-compliant reasoning paths without sacrificing detection accuracy. Among the individual components, removing the planning stage causes the most catastrophic drop in strict F1 scores across all splits. Eliminating audio understanding also substantially lowers performance, indicating that direct tool calling alone is insufficient without physical observations. Furthermore, excluding description evidence severely degrades the parsing rate on C2 to 23.69%, confirming that structured evidence aggregation is essential for maintaining stable LLM reasoning in overlapping environments.

To evaluate trajectory precision, Table 5 assesses the fine-grained localization performance of ToolDF following standard DCASE evaluation metrics (Turpault et al., 2019). The framework demonstrates stable localization capabilities across all subsets, yielding macro F1 scores between 93.64% and 99.97% in single-type domains for both segment-level and event-level metrics. In hybrid mixed environments, ToolDF maintains consistent localization performance, achieving event-level macro-F1 scores of 94.24% on temporal transitions (C1) and 87.62% on hybrid mixtures (C3). These fine-grained results confirm that ToolDF is fully capable of going beyond clip-level binary classification to localize the specific time segments and acoustic sources that constitute synthetic evidence.

7 Conclusion

We presented ToolDF, an innovative tool-integrated reasoning framework for mixed-authenticity audio deepfake detection. By leveraging an Audio Large Language Model (ALLM) as an intelligent orchestrator, ToolDF dynamically analyzes audio structures, adaptively routes heterogeneous inputs to specialized expert detectors, and seamlessly aggregates component-level evidence into an interpretable clip-level verdict. Extensive experiments on our newly constructed mixed-authenticity benchmark demonstrate that ToolDF significantly outperforms conventional monolithic baselines and fixed pipelines, while simultaneously establishing robust, fine-grained localization capabilities for real-world audio manipulation.

Limitations

ToolDF depends on external tools, including source separation modules and domain-specific deepfake detectors. Therefore, errors from source separation or detector misclassification can propagate to the final decision, and the overall performance is bounded by the reliability and coverage of the underlying expert models.

In addition, our benchmark is constructed by composing existing public datasets. While this enables controlled evaluation with component-level annotations, it may not fully reflect the diversity, editing artifacts, source interactions, and distributional complexity of real-world manipulated media. The composition rules used for temporal transitions and acoustic overlaps may also be simpler than naturally occurring or adversarially edited audio.

Finally, ToolDF currently focuses on four acoustic domains: speech, singing, music, and environmental sound. Extending the framework to broader audio events, open-set acoustic domains, and unseen manipulation types remains future work.

References

Darius Afchar, Gabriel Meseguer-Brocal, and Romain Hennequin. 2025. Ai-generated music detection and its challenges. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Andrea Agostinelli, Timo I Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, and 1 others. 2023. Musiclm: Generating music from text. *arXiv preprint arXiv:2301.11325*.

Arun Babu, Changhan Wang, Andros Tjandra, Kushal Lakhotia, Qiantong Xu, Naman Goyal, Kritika Singh, Patrick von Platen, Yatharth Saraf, Juan Pino, Alexei Baevski, Alexis Conneau, and Michael Auli. 2022. Xls-r: Self-supervised cross-lingual speech representation learning at scale. In *Interspeech 2022*, pages 2278–2282.

Luca Comanducci, Paolo Bestagini, and Stefano Tubaro. 2025. Fakemusiccaps: A dataset for detection and attribution of synthetic music generated via text-to-music models. *Journal of Imaging*, 11(7):242.

Hao Gu, Jiangyan Yi, Chenglong Wang, Jianhua Tao, Zheng Lian, Jiayi He, Yong Ren, Yujie Chen, and Zhengqi Wen. 2025. Allm4add: Unlocking the capabilities of audio large language models for audio deepfake detection. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 11736–11745.

Xiaoxuan Guo, Yuankun Xie, Haonan Cheng, Jiayi Zhou, Jian Liu, Hengyan Huang, Long Ye, and Qin Zhang. 2026. Towards explicit acoustic evidence perception in audio llms for speech deepfake detection. *arXiv preprint arXiv:2601.23066*.

Jee-weon Jung, Hee-Soo Heo, Hemlata Tak, Hye-jin Shim, Joon Son Chung, Bong-Jin Lee, Ha-Jin Yu, and Nicholas Evans. 2022. Aasist: Audio anti-spoofing using integrated spectro-temporal graph attention networks. In *ICASSP 2022-2022 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 6367–6371. IEEE.

Taewoo Kim, Guisik Kim, Choongsang Cho, and Young Han Lee. 2025. Naturalness-aware curriculum learning with dynamic temperature for speech deepfake detection. In *Interspeech 2025*, pages 5318–5322.

Yuang Li, Min Zhang, Mengxin Ren, Xiaosong Qiao, Miaomiao Ma, Daimeng Wei, and Hao Yang. 2024. Cross-domain audio deepfake detection: Dataset and analysis. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4977–4983.

Andreas Nautsch, Xin Wang, Nicholas Evans, Tomi H Kinnunen, Ville Vestman, Massimiliano Todisco, Héctor Delgado, Md Sahidullah, Junichi Yamagishi, and Kong Aik Lee. 2021. Asvspoof 2019: Spoofing countermeasures for the detection of synthesized, converted and replayed speech. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 3(2):252–265.

Cheng Qian, Emre Can Acikgoz, Qi He, Hongru Wang, Xiusi Chen, Dilek Hakkani-Tur, Gokhan Tur, and Heng Ji. 2026. Toolrl: Reward is all tool learning needs. *Advances in Neural Information Processing Systems*, 38:105523–105553.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, and 1 others. 2024. Toolllm: Facilitating large language models to master 16000+ real-world apis. In *International Conference on Learning Representations*, volume 2024, pages 9695–9717.

Eros Rosello, Alejandro Gomez-Alanis, Angel M. Gomez, and Antonio Peinado. 2023. A conformer-based classifier for variable-length utterance processing in anti-spoofing. In *Interspeech 2023*, pages 5281–5285.

Simon Rouard, Francisco Massa, and Alexandre Défossez. 2023. Hybrid transformers for music source separation. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Andrew Rouditchenko, Saurabhchand Bhati, Edson Araujo, Samuel Thomas, Hilde Kuehne, Rogerio Feris, and James Glass. 2025. Omni-r1: Do you really need audio to fine-tune your audio llm? *arXiv preprint arXiv:2505.09439*.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *Advances in neural information processing systems*, 36:68539–68551.

Hemlata Tak, Massimiliano Todisco, Xin Wang, Jee weon Jung, Junichi Yamagishi, and Nicholas Evans. 2022. Automatic speaker verification spoofing and deepfake detection using wav2vec 2.0 and data augmentation. In *The Speaker and Language Recognition Workshop (Odyssey 2022)*, pages 112–119.

Nicolas Turpault, Romain Serizel, Ankit Parag Shah, and Justin Salamon. 2019. Sound event detection in domestic environments with weakly labeled data and soundscape synthesis. In *Workshop on Detection and Classification of Acoustic Scenes and Events*.

Yuankun Xie, Ruibo Fu, Xiaopeng Wang, Zhiyong Wang, Songjun Cao, Long Ma, Haonan Cheng, and Long Ye. 2026a. Detect all-type deepfake audio: Wavelet prompt tuning for enhanced auditory perception. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 35922–35930.

Yuankun Xie, Xiaoxuan Guo, Jiayi Zhou, Tao Wang, Jian Liu, Ruibo Fu, Xiaopeng Wang, Haonan Cheng, and Long Ye. 2026b. Interpretable all-type audio deepfake detection with audio llms via frequency-time reinforcement learning. *arXiv preprint arXiv:2601.02983*.

Yuankun Xie, Yi Lu, Ruibo Fu, Zhengqi Wen, Zhiyong Wang, Jianhua Tao, Xin Qi, Xiaopeng Wang, Yukun Liu, Haonan Cheng, and 1 others. 2025. The codecfake dataset and countermeasures for the universally detection of deepfake audio. *IEEE Transactions on Audio, Speech and Language Processing*, 33:386–400.

Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, Bin Zhang, Xiong Wang, Yunfei Chu, and Junyang Lin. 2025. Qwen2.5-omni technical report. *arXiv preprint arXiv:2503.20215*.

Junichi Yamagishi, Xin Wang, Massimiliano Todisco, Md Sahidullah, Jose Patino, Andreas Nautsch, Xuechen Liu, Kong Aik Lee, Tomi Kinnunen, Nicholas Evans, and Héctor Delgado. 2021. Asvspoof 2021: accelerating progress in spoofed and deepfake speech detection. In *2021 Edition of the Automatic Speaker Verification and Spoofing Countermeasures Challenge*, pages 47–54.

Han Yin, Yang Xiao, Rohan Kumar Das, Jisheng Bai, Haohe Liu, Wenwu Wang, and Mark D Plumbley. 2025. EnvSDD: Benchmarking Environmental Sound Deepfake Detection. In *Interspeech 2025*, pages 201–205.

Yongyi Zang, Jiatong Shi, You Zhang, Ryuichi Yamamoto, Jionghao Han, Yuxun Tang, Shengyuan Xu, Wenxiao Zhao, Jing Guo, Tomoki Toda, and Zhiyao Duan. 2024a. CtrSVDD: A Benchmark Dataset and Baseline Analysis for Controlled Singing Voice Deepfake Detection. In *Interspeech 2024*, pages 4783–4787.

Yongyi Zang, You Zhang, Mojtaba Heydari, and Zhiyao Duan. 2024b. Singfake: Singing voice deepfake detection. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 12156–12160. IEEE.

Lin Zhang, Xin Wang, Erica Cooper, Nicholas Evans, and Junichi Yamagishi. 2022. The partialspoof database and countermeasures for the detection of short fake speech segments embedded in an utterance. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:813–825.

Xueping Zhang, Yechen Wang, Linxi Li, Liwei Jin, and Ming Li. 2026. Compspoof: A dataset and joint learning framework for component-level audio anti-spoofing countermeasures. In *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 18067–18071. IEEE.

A Mixed-Authenticity Benchmark Construction

This appendix provides additional details on the construction pipeline and statistics of the Mixed-Authenticity Audio Deepfake Detection Benchmark. We first describe how single-source audio clips are converted into structured component annotations, and then explain how these annotations are used to synthesize composite mixtures for the C1, C2, and C3 settings.

A.1 Audio Caption and Component Annotation

Before constructing composite mixtures, we annotate each single-source audio clip with structured component-level metadata. Each component is associated with a content type, a support region, and an authenticity label. The content type is selected from speech, singing, music, and environmental sound. The support region specifies the temporal span in which the component is active, and the authenticity label indicates whether the component is real or fake.

Audio Captioning Prompt Configuration

System Prompt
You are an audio captioner. Output exactly one short English sentence (about 10–20 words) that captures only the most essential audible traits of the given `{category}` clip. Match the style and length of this example: `{example}`.

Rules by category:

- `speech`: describe gender, delivery tone, speaking pace, and pitch range. Do not describe the acoustic environment, such as quietness, noise, or reverberation.
- `singing`: describe gender and melody or singing style. Do not mention instruments, accompaniment, or background music.
- `music`: describe the main instruments and mood or tempo.
- `sound`: describe the main environmental sound source or sources.

Describe only what is clearly audible. Do not invent details, and do not mention background elements that are not present. Do not mention deepfake, authenticity, AI, models, or tools. Output only the sentence, with no prefix, label, quotation marks, or extra text.

User Prompt
<audio> Caption this `{category}` clip.

Figure 3: Prompt configuration used for Qwen3-Omni-30B-A3B-Captioner-based single-source audio caption generation. The prompt constrains captions to observable acoustic properties and excludes authenticity-related information to avoid label leakage.

We also generate a concise single-source audio caption for each component using Qwen3-Omni-30B-A3B-Captioner[2]. The caption describes only observable acoustic properties, such as vocal delivery, singing style, instrumentation, or environmental sound sources, depending on the component type. Importantly, the captioning prompt explicitly prevents the model from mentioning authenticity-related information, including whether the audio is real, fake, synthetic, AI-generated, or produced by a model. The detailed prompt configuration and category-specific rules are provided in Table 3. This design avoids label leakage while preserving the acoustic information needed for trajectory construction.

After composite mixtures are constructed, the captions are combined with the component metadata to form the `<audio_understanding>` block of the supervised ToolDF trajectory. Thus, the model learns not only the final clip-level authenticity label, but also the intermediate acoustic structure needed for tool selection and evidence aggregation.

Boundary Annotation. For vocal domains, including speech and singing, we estimate active regions using WebRTC-VAD[3]. This removes prolonged silence and yields localized support regions for vocal components. For non-vocal domains, including music and environmental sound, we use the full clip duration as the valid support region, since these signals typically function as continuous background components in the constructed mixtures.

[2] https://huggingface.co/Qwen/Qwen3-Omni-30B-A3B-Captioner

[3] https://github.com/wiseman/py-webrtcvad

A.2 Composite Mixture Construction

Using the annotated single-source component pools, we construct three types of composite mixed-authenticity audio: temporal transitions, acoustic overlaps, and hybrid mixtures. Each composite example is created by combining components from different acoustic domains while preserving their component-level support regions and authenticity labels. The clip-level label is then assigned according to the early-fail rule: a composite clip is labeled fake if at least one of its constituent components is fake, and real otherwise.

Temporal Transition (C1). The C1 setting models temporal transitions between two vocal domains. We construct each example by sequentially concatenating a speech clip and a singing clip. Both transition orders are considered: Speech $\rightarrow$ Singing and Singing $\rightarrow$ Speech. The support region of each component is updated according to its position in the concatenated sequence, so that the resulting trajectory can identify which temporal segment corresponds to speech and which corresponds to singing.

Split	Acoustic Configuration	train (R/F)	dev (R/F)	eval (R/F)	Total
C1 (Temporal)	Speech → Singing	645 / 12,045	637 / 11,785	1,838 / 33,778	60,728
	Singing → Speech	645 / 12,045	637 / 11,785	1,839 / 33,781	60,732
	C1 Subtotal	**1,290 / 24,090**	**1,274 / 23,570**	**3,677 / 67,559**	**121,460**
C2 (Overlap)	Speech + Music	159 / 3,011	170 / 2,155	170 / 2,944	8,609
	Singing + Music	657 / 9,624	361 / 3,734	354 / 3,803	18,533
	Speech + Environmental Sound	1,155 / 21,055	1,108 / 12,952	867 / 16,367	53,504
	Singing + Environmental Sound	5,403 / 68,719	2,867 / 22,783	1,630 / 20,904	122,306
	C2 Subtotal	**7,374 / 102,409**	**4,506 / 41,624**	**3,021 / 44,018**	**202,952**
C3 (Hybrid)	(Speech → Singing) + Music	33 / 1,385	40 / 1,157	154 / 825	3,594
	(Singing → Speech) + Music	30 / 1,461	35 / 1,187	177 / 808	3,698
	(Speech → Singing) + EnvSound	293 / 10,979	291 / 7,524	758 / 4,303	24,148
	(Singing → Speech) + EnvSound	289 / 10,910	271 / 7,519	749 / 4,310	24,048
	C3 Subtotal	**645 / 24,735**	**637 / 17,387**	**1,838 / 10,246**	**55,488**
Total	**Combined Composite Pool**	**9,309 / 151,234**	**6,417 / 82,581**	**8,536 / 121,823**	**379,900**

Table 6: Detailed statistics of the composite-type configurations in the mixed-authenticity audio deepfake detection benchmark. C1 denotes temporal speech–singing transitions, C2 denotes acoustic overlaps between vocal and non-vocal components, and C3 denotes hybrid mixtures combining temporal transitions and acoustic overlaps.

To cover different mixed-authenticity patterns, we construct four authenticity combinations: real-to-real ($R \rightarrow R$), real-to-fake ($R \rightarrow F$), fake-to-real ($F \rightarrow R$), and fake-to-fake ($F \rightarrow F$). The first three combinations are sampled with the same budget to ensure balanced coverage of real and partially fake transitions. The fake-to-fake subset is then filled using the remaining available fake source pools. This design allows C1 to evaluate whether a model can detect localized manipulations across temporal changes in vocal content.

Acoustic Overlap (C2). The C2 setting models overlapping foreground and background sources. We mix a foreground vocal component with a non-vocal background component. The foreground component is either speech or singing, and the background component is either music or environmental sound. This results in four sub-configurations: Speech + Music, Speech + Environmental Sound, Singing + Music, and Singing + Environmental Sound.

For each mixture, the vocal component retains its localized support region, while the background component is treated as active over the full mixture duration. The resulting example therefore contains source-level overlap between a foreground vocal stream and a background non-vocal stream. The number of examples in each sub-configuration follows the available source-pool sizes, which naturally yields larger subsets for singing and environmental sound.

Hybrid Mixture. (C3) The C3 setting combines the two previous composition types. We first construct a C1-style temporal transition between speech and singing, and then overlay the resulting vocal sequence with a non-vocal background component, either music or environmental sound. This produces hybrid mixtures that contain both temporal changes in the foreground vocal content and acoustic overlap with a background source.

As in C2, the background component is treated as active over the full mixture duration, while the speech and singing components retain their updated temporal support regions from the transition sequence. C3 is therefore the most challenging setting, requiring the model to reason over both temporal segmentation and source-level separation before aggregating component-level authenticity evidence.

A.3 Dataset Statistics

The complete benchmark contains 973,649 examples, including 379,900 composite examples from the C1, C2, and C3 settings. Table 6 reports the detailed composite-type statistics across train, development, and evaluation splits. The composite pool is fake-heavy because a clip is labeled fake whenever at least one constituent component is fake; the relatively higher real ratio in the C3 evaluation split results from basename-level resplitting

ToolDF Prompt Configuration

System Prompt
You are an expert in audio deepfake detection. You MUST use the provided tools to judge authenticity — never decide by yourself, and never fabricate tool results.

Within `<think>`, write `<audio_understanding>` (segment-by-segment analysis with timestamps) and `<plan>` (which detectors to call on which segments, and whether `source_separator` is needed).

After tool results return, write `<description>` summarizing each tool's prediction and score, then `<answer>` with a single word: `real` or `fake`. Any fake detection means the overall answer is `fake`.

User Prompt
<audio> Is this audio real or fake?

Figure 4: System prompt configuration used during supervised fine-tuning (SFT) and inference to enforce the Tool-Integrated Reasoning protocol for the ToolDF orchestrator.

of FakeMusicCaps to prevent background-track leakage.

B System Prompt for ToolDF

To ensure the reproducibility of the Tool-Integrated Reasoning (TIR) trajectory, we provide the exact system prompt injected into the ALLM orchestrator during both supervised fine-tuning (SFT) and inference. As specified in Section 3.3, this prompt enforces the structured generation format and prevents the model from fabricating tool results or introducing hallucinated reasoning paths.