CodecSep: Prompt-Driven Universal Sound Separation on Neural Audio Codec Latents

Abstract

Text-guided sound separation supports flexible audio editing across media and assistive applications, but existing models like AudioSep are too compute-heavy for edge deployment. Neural Audio Codec–based models such as CodecFormer and SDCodec are compute efficient but limited to fixed-class separation. We introduce CodecSep, a NAC-based model for universal, text-driven sound separation designed for on-device and edge deployment in low-latency, codec-mediated settings, as supported by its favorable compute and memory footprint. CodecSep combines DAC compression with a lightweight transformer masker modulated by CLAP-derived FiLM parameters. A key empirical finding underlying this design is that the latent space of modern neural audio codecs is already sufficiently structured: source-specific information is partially disentangled in the NAC representation, allowing effective source extraction via masking alone, without explicit re-encoding. Qualitative analysis of the codec representations further supports this view, showing that the latent space preserves source-dependent structure and is sufficiently disentangled to enable effective source separation. Across six open-domain benchmarks under matched training and prompting protocols, CodecSep surpasses AudioSep in separation fidelity (SI-SDR) while remaining competitive in perceptual quality (ViSQOL) and matching or exceeding fixed-stem baselines (TDANet, Sudo rm-rf, CodecFormer, SDCodec). In code-stream deployments, it requires just 1.35 GMACs end-to-end—$\sim54\times$ less compute ($25\times$ architecture-only) than spectrogram-domain separators like AudioSep—while remaining fully bitstream-compatible.

1 Introduction

We propose CodecSep, a text-conditioned universal sound separation (USS) framework that marries the interpretability of prompt-driven extraction with the efficiency of neural audio codecs (NACs). To our knowledge, CodecSep is the first system to bridge NACs with USS: it conditions a transformer *masker* on CLAP text embeddings Wu et al. (2023) via Feature-wise Linear Modulation (FiLM) Perez et al. (2018), and performs separation directly in the codec encoder latent space. This design introduces semantic control while retaining the compact computational footprint of codec representations, making prompt-driven separation particularly well-suited to low-latency, codec-mediated deployment, including edge and potentially on-device settings.

Flexible, real-time separation on bandwidth- or compute-constrained platforms remains challenging. Classic models disentangle sources from complex mixtures Vincent et al. (2018) but are often domain-specific (e.g., speech/music) and heavy. Recent text-guided systems like AudioSep Liu et al. (2024) extend encoder–masker–decoder designs (e.g., Conv-TasNet-style Luo & Mesgarani (2019)) by injecting semantics from BERT/CLAP through FiLM layers Devlin et al. (2019); Wu et al. (2023); Perez et al. (2018). However, spectrogram/waveform-domain separators trained with SI-SDR-style losses Luo & Mesgarani (2019); Le Roux et al. (2019) are compute-intensive and sensitive to compression artifacts, often pushing inference to the cloud.

NACs such as SoundStream, Encodec, and DAC Zeghidour et al. (2022); Défossez et al. (2022); Kumar et al. (2023) compress audio to discrete tokens with Residual Vector Quantization (RVQ), providing compact,

perceptually aligned latents useful for generation and conditioned synthesis Borsos et al. (2023); Wang et al. (2023; 2024); Du et al. (2024). Prior codec–separation hybrids (CodecFormer Yip et al. (2024b), SDCodec Bie et al. (2024)) are lightweight and high-fidelity but target fixed stems (e.g., speech separation or speech vs. music vs. SFX); extending them to open-domain, prompt-conditioned USS is non-trivial (cf. §2, para. 3).

CodecSep adopts a frozen DAC encoder–decoder backbone and inserts a FiLM-conditioned transformer masker that predicts a soft mask over codec latents. CLAP-derived text embeddings Wu et al. (2023) are mapped to per-layer FiLM parameters, modulating the masker's intermediate activations to align the selected latent subspace with the query semantics. Operating on compact codec features cuts memory traffic and MACs compared to spectrogram-domain pipelines, while preserving the codec's inductive biases (periodicity, timbre, transients). In doing so, CodecSep delivers interpretable, prompt-guided separation with markedly lower compute without sacrificing separation fidelity. Crucially, conditioning via text embeddings enables open-vocabulary operation of NAC-based separation.

We evaluate CodecSep across: (i) *in-domain text-guided separation* on dnr-v2 Petermann et al. (2022); (ii) *cross-domain generalization* on five open-domain corpora (AudioCaps Kim et al. (2019), ESC-50 Piczak, Clotho-v2 Drossos et al. (2020), AudioSet-eval Gemmeke et al. (2017), VGGSound Chen et al. (2020a)); (iii) *three prompt granularities* (fixed-stem, generic three-stem, and fine-grained SFX); (iv) *paraphrase robustness*; (v) an *architectural ablation* comparing decoder-style generation to our transformer masker; (vi) a *bandwidth-scaling study* extending CodecSep to higher sampling rates; and (vii) *compute benchmarking* on small GPUs. We benchmark against the SOTA text-guided baseline, AudioSep Liu et al. (2024), under matched data and prompt protocols. Across all benchmarks, the strong performance of the lightweight masker indicates that NAC latents are already sufficiently source-structured: simple, text-conditioned masking is enough to isolate sources without explicit re-encoding. CodecSep consistently surpasses AudioSep in SI-SDR while remaining competitive in ViSQOL, and it degrades more gracefully under prompt paraphrasing. In deployment-typical *code-stream* settings, where audio is already exchanged as codec bitstreams, CodecSep runs at just 1.35 GMACs end-to-end—$\sim$54$\times$ less compute than AudioSep in this regime (and $\sim$25$\times$ lower separator-only compute)—while remaining fully compatible with bitstream interfaces.

2 Related Work

Classical sound separation systems frequently adopt an encoder–masker–decoder design in which an encoder produces STFT-like latents, a masker predicts source-specific masks, and a decoder reconstructs waveforms. Representative models include DPTNet Chen et al. (2020b), SepFormer Subakan et al. (2021), and TDANet Li et al. (2023), the last introducing a top-down attention scheme that blends global and local attention to capture multi-scale acoustic structure. Beyond masking pipelines, several works generate waveforms directly in the time domain (Wave-UNet Stoller et al. (2018), Demucs Défossez et al. (2019); Défossez et al. (2021)) or operate fully in the complex STFT domain with joint magnitude–phase modeling (MM-DenseLSTM Takahashi et al. (2018), DCCRN Hu et al. (2020), Spleeter Hennequin et al. (2020)), underscoring the breadth of design choices.

Moving from domain-specific separation to *universal* sound separation (USS), supervised systems typically rely on Permutation Invariant Training (PIT) Yu et al. (2017); Kavalerov et al. (2019), while unsupervised methods such as MixIT Wisdom et al. (2020) learn directly from mixtures. Both paradigms assume a fixed maximum number of sources and output all estimates indiscriminately, requiring a post-hoc identification step (cf. Appendix A for detailed failure modes). A recent PIT-trained USS model is Sudo rm-rf Tzinis et al. (2022a), a parameter-efficient time-domain separator that is based on ConvTasNet-style encoder–masker–decoder architecture with adaptive encoder/decoder modules. It downsamples input waveform to STFT-like latents before separation and is PIT-trained to separate mixtures with up to four sources. Query-Guided Sound Separation (QSS) addresses this limitation of PIT or MixIT models by conditioning extraction on external queries—visual cues, audio, class labels, or text. Text queries are compact, expressive, and capture high-level semantics without requiring additional reference signals. AudioSep Liu et al. (2024) follows this direction by injecting BERT Devlin et al. (2019) or CLAP Wu et al. (2023) embeddings via FiLM Perez et al. (2018) at intermediate masker layers to steer separation toward the query source. BiModalSS Mahmud et al. (2024) extends AudioSep with attention-based conditioning and more efficient training strategies. FiLM-conditioned

Figure 1: An overview of CodecSep. (Left) The full pipeline for text-guided USS. (Right) The integration of text conditioning into intermediate layers of transformer masker via FiLM layers.

variants of Sudo rm-rf Tzinis et al. (2022b; 2023) have also been explored for class-guided separation using one-hot or multi-hot conditioning vectors; however, such label-based conditioning does not extend naturally to open-domain text queries and cannot handle unseen classes.

Neural audio codecs (NACs) have recently been integrated into separation pipelines to improve efficiency. CodecFormer separates directly in DAC Kumar et al. (2023) latent space with a transformer trained using negative SI-SDR, and CodecFormer-EL Yip et al. (2024a) adds an embedding-level objective to align separator outputs with encoder latents. SDCodec Bie et al. (2024) embeds separation inside the codec by assigning dedicated RVQ branches to speech, music, and SFX and summing their codes to form mixture representations. However, neither design trivially extends to open-domain, prompt-conditioned USS: SDCodec's hardwired, per-stem RVQ branches do not scale to open vocabularies (adding branches explodes parameters, while a mixture-invariant reformulation collapses into "three generic codecs" with no stem-specific RVQ specialization). MixIT-style training of CodecFormer still presupposes a maximum number of sources, conflicting with universal separation.

3 Method

CodecSep adapts the 16 kHz DAC Kumar et al. (2023) codec backbone for text-driven universal sound separation (USS). We use a transformer masker that estimates soft masks over codec latents and inject text conditioning via Feature-wise Linear Modulation (FiLM) Perez et al. (2018) using CLAP text embeddings Wu et al. (2023). FiLM is applied to intermediate transformer activations so the query semantics steer separation. Figure 1 (Left) shows the overall text-guided pipeline; Figure 1 (Right) highlights the FiLM-conditioned masker. Compared to STFT-domain AudioSep, operating in compact codec latents yields markedly lower compute and is amenable to edge deployment.

3.1 Task Formulation

We consider a mono mixture as $x(t) = \sum_{s \in \mathcal{S}} y_s(t)$, where $x(t)$ is the observed waveform, $\mathcal{S}$ is the (unbounded) set of source classes/instances present, and $y_s(t)$ is the waveform of source s. Given a natural-language query τ (e.g., "dog barking", "speech and music"), the goal is to recover the waveform of the source consistent with the query.

In spectrogram-domain text-guided separation systems such as AudioSep, this takes the form

$$x(t) \xrightarrow{\text{STFT}} X \in \mathbb{C}^{F \times T_{spec}} \xrightarrow{Spec(X, e_\tau)} \tilde{Y}_s = |\hat{M}_s| \odot |X| \exp\left(\angle X + \angle \hat{M}_s\right) \xrightarrow{\text{ISTFT}} \tilde{y}_s(t), \quad (1)$$

where $Spec(\cdot, e_\tau)$ is a FiLM-conditioned masker that predicts a magnitude mask $|\hat{M}_s| \in [0, 1]^{F \times T_{spec}}$ (F: frequency bins, T_{spec}: spectrogram frames) and a phase residual $\angle \hat{M}_s$ given the complex STFT X of audio $x(t)$ and text-embedding e_τ.

In CodecSep, the task is posed directly in the NAC latent domain:

$$x(t) \xrightarrow[\text{DAC}]{Enc(\cdot)} Z \in \mathbb{R}^{d \times T} \xrightarrow{Mask(Z, e_\tau)} \tilde{Z}_s = M_s \odot Z \xrightarrow[\text{DAC}]{Dec(\cdot)} \tilde{y}_s(t), \qquad (2)$$

with frozen DAC backbone $Enc(\cdot), Dec(\cdot)$ and a FiLM-conditioned transformer masker $Mask(\cdot, e_\tau)$ that estimates mask $M_s \in [0,1]^{d \times T}$ applied element-wise to DAC latent Z.

Thus, for an input clip producing T latent frames, the separator receives $Z \in \mathbb{R}^{d \times T}$, predicts a same-shape mask $M_s \in [0,1]^{d \times T}$, forms the masked latent $\tilde{Z}_s \in \mathbb{R}^{d \times T}$, and decodes it to the waveform estimate $\tilde{y}_s(t)$. In the main system, the separator therefore acts as a selection mechanism over codec latents rather than a source generator.

3.2 Model Architecture

3.2.1 Descript Audio Codec (DAC) Backbone

We use DAC Kumar et al. (2023) as encoder–decoder. Following Encodec/SoundStream, DAC uses fully-convolutional encoder/decoder with the periodic Snake activation $x + \sin^2 x$ (replacing LeakyReLU) to bias periodic audio modeling. Residual vector quantization (RVQ) compresses encoder outputs with factorized codes and ℓ_2-normalized codebooks. For a 1 s audio $x(t)$ at F_s=24 kHz compressed to R=6000 bps, the encoder $Enc(\cdot)$ downsamples by M=320 to $T=F_s/M$=75 frames of latents $Z=[z_t \in \mathbb{R}^d]_{t=1}^T$ (d: channel width) with $r=R/T$=80 bits/frame. RVQ allocates $r_i=r/N_q$=10 bits across N_q=8 codebooks (size 2^{10}=1024). Given Z, $Quant(\cdot)$ yields discrete codes $A=[a_t \in [1024]^8]$, which map to embeddings $e_t=\sum_{i=1}^8 e_t^i$; $Dec(\cdot)$ upsamples $E=[e_t]$ back to waveform $y(t)$.

3.2.2 Why NAC latents vs. spectrograms

Operating on NAC latents Z slashes dimensionality while preserving perceptual factors. For 1 s audio at 32 kHz, complex STFT with N=1024 and hop size M=320 samples has $T_{\text{spec}} \approx 100$ frames and $F = 2 \times 1024$ (Re+Im) scalars per frame, so $F \cdot T_{spec} \approx 204{,}800$. A 16 kHz DAC with width d=64 and the same M yields $T \approx 50$ and $d \cdot T$=64 × 50=3,200 ($\sim 64\times$ smaller), shrinking $Q/K/V$ and MLP sizes and easing self-attention. Similarly, for 32 kHz NACs like EnCodec, $T \approx 50$ with d=128, so attention/MLPs still operate on $\sim 32\times$ smaller latents than complex STFTs. Crucially, $Enc(\cdot)$ organizes Z on a discriminative, perceptually aligned manifold, making selection (masking) easier than representation learning from raw X. In spectrogram systems $Spec(.)$, the separator must first learn a high-level latent from X (via CNN/UNet) and then separate, coupling abstraction and masking and inflating parameters/compute. Waveform separators $Wave(.)$ such as Sudo rm-rf Tzinis et al. (2022a;b; 2023) likewise downsample the waveform into STFT-like intermediate latents via 1D convolutions before encoding, resulting in latents with similar dimensionality and thus inheriting the same challenges as spectrogram systems.

3.2.3 FiLM-conditioned Transformer Masker

Leveraging the codec prior. Because the DAC codec induces a strong semantic prior in its latent space via residual vector quantization (RVQ) and perceptual/adversarial training, we mask the codec latents rather than generate sources from scratch as in CodecFormer. RVQ creates a coarse-to-fine hierarchy in $Z=Enc(x) \in \mathbb{R}^{d \times T}$: early stages capture coarse structure (e.g., low-frequency content, timbre), while later stages refine residual detail (e.g., high-frequency components, transients) Wang et al. (2023). We exploit this organization with a FiLM-conditioned transformer masker that predicts a soft mask $M_s \in [0,1]^{d \times T}$ and applies it element-wise, yielding source latent estimate $\tilde{Z}_s = M_s \odot Z$.

Masking, not generating. In contrast to learning a generator $Gen(Z, e_\tau) : Z \to \tilde{Z}_s$ as in CodecFormer, learning a $mask\ Mask(Z, e_\tau) : Z \to M_s$ on the compact, semantically organized codec manifold both exploits the codec prior more effectively and yields a more stable optimization that converges faster. Moreover, masking in the denoised, low-dimensional codec space is fundamentally easier than masking in the high-dimensional, noisy spectrogram domain. This selection-centric design (i) constrains learning to modulation of existing

latent content, (ii) avoids hallucination and reduces leakage because no new signals are synthesized, and (iii) preserves long-horizon structure (periodicity, timbre, transients) already organized by the codec, yielding stable, low-artifact separations.

Architecture and dimensional interface. Concretely, passing $x(t)$ through the frozen DAC encoder yields codec latents $Z \in \mathbb{R}^{d \times T}$ (d: codec channel width, T: latent frames). The masker $Mask(\cdot)$ operates in this latent space to predict an element-wise mask that selects the target source.

Since the codec latent dimensionality d can differ from the transformer width, we introduce lightweight channel projections to interface between the two. Specifically, the codec latents are first mapped to the transformer width d_t using a pointwise convolution,

$$Z' = \text{Conv}(Z), \quad Z' \in \mathbb{R}^{d_t \times T}, \tag{3}$$

after which all transformer operations are performed in this space.

We adopt a CodecFormer-style transformer with L=16 layers, width d_t=256, and Snake activations. Given the natural-language query τ, we compute a CLAP text embedding $e_\tau \in \mathbb{R}^{d_t}$. A lightweight query network $query(.)$—implemented as a single linear layer—maps e_τ to per-layer FiLM parameters $(\gamma^l, \beta^l) \in \mathbb{R}^{d_t}$ for $l \in \{2, \ldots, L-1\}$, applied channel-wise to intermediate activations $H^l \in \mathbb{R}^{d_t \times T}$:

$$\tilde{H}^l = \text{FiLM}(H^l; \gamma^l, \beta^l) = \gamma^l \odot H^l + \beta^l. \tag{4}$$

Thus, FiLM is injected at the intermediate transformer layers $l = 2, \ldots, L - 1$, while the first and final transformer layers remain unmodulated.

The final transformer output H^L is then passed through a convolutional mask head consisting of a 1D convolution followed by a pointwise (1×1) convolution. Together, these layers map the transformer output from width d_t back to the codec latent dimensionality d and produce the prompt-conditioned mask $M_s \in [0,1]^{d \times T}$. The mask is then applied element-wise to the codec latents, yielding $\tilde{Z}_s = M_s \odot Z$. Finally, the waveform estimate is obtained with the frozen codec decoder as $\tilde{y}_s = Dec(\tilde{Z}_s)$, bypassing RVQ lookup.

Temporal resolution is preserved throughout, and all projections are channel-wise operations. This design explicitly handles the dimensionality mismatch between codec latents and transformer representations while maintaining a direct, shape-aligned masking operation in the original codec latent space.

Why FiLM inside the masker. Placing FiLM in the masker (rather than in Enc/Dec) directly targets the selection step while preserving the codec manifold. We further adopt a post-LN FiLM design: modulation is applied after the transformer sublayer activations, rather than through the normalization itself. In our setting, this is desirable because the conditioning behaves as an explicit channel-wise feature gate, allowing the masker to suppress interference and emphasize target-relevant latent components without perturbing the backbone normalization statistics. This design introduces minimal overhead (two vectors per layer) and maintains a non-iterative, single-pass conditioning mechanism, enabling low-latency inference in both edge and server settings.

Why Post-LN FiLM instead of AdaLN. We apply FiLM after the transformer sublayer rather than conditioning the normalization itself (AdaLN-style), following prior text-guided separation work such as AudioSep. Our motivation is task-specific. CodecSep performs *masking-based source selection*, not generative synthesis, so the conditioning mechanism must act as directly as possible on the features that determine whether a latent component is preserved or suppressed. In this setting, Post-LN FiLM is well aligned with the separation objective because it modulates the hidden activations *after* they have been computed, through channel-wise affine conditioning. Operationally, this makes the text signal behave like an explicit feature gate: for a given query, it can directly attenuate channels carrying interfering content and emphasize channels carrying target-relevant structure, while leaving the backbone normalization statistics unchanged.

By contrast, AdaLN injects conditioning into the normalization transform itself. This is highly effective in generative architectures such as DiT Peebles & Xie (2023), where conditioning is intended to steer representation formation in a broad and distributed manner. However, for source separation—especially in

our lightweight low-GMAC regime—the goal is not broad conditional generation but sharp source-selective suppression. In that regime, distributing conditioning through the normalization pathway can lead to more diffuse modulation of the representation, whereas Post-LN FiLM provides a simpler and more targeted mechanism for source-conditioned masking. We therefore adopt Post-LN FiLM as the conditioning scheme that is more naturally matched to explicit latent selection in codec-space separation.

This interpretation is also consistent with our qualitative analysis in § 4.1.4, where the learned masks appear predominantly channel-wise rather than strongly time-localized, which is consistent with the view that separation is achieved mainly through FiLM-conditioned feature reweighting within the masker.

FiLM parameterization and training stability. Following AudioSep, we adopt a simplified FiLM parameterization where only the bias term is learned, while the scale is fixed. Concretely, we set $\gamma^l = \mathbf{1}$ for all layers and learn only β^l, so that FiLM reduces to an additive modulation:

$$\tilde{H}^l = H^l + \beta^l. \tag{5}$$

The FiLM layers are implemented as linear projections from the text embedding, with weights initialized using Xavier uniform initialization Glorot & Bengio (2010) and biases initialized to zero. This ensures that $\beta^l = \mathbf{0}$ at initialization, so the conditioned transformer initially behaves identically to an unmodulated masker.

This design improves training stability by avoiding uncontrolled multiplicative scaling of hidden activations, which can be particularly problematic in Post-LN architectures. Moreover, in the context of masking-based separation, additive modulation is sufficient to introduce source-selective cues while preserving the structure of the codec latent manifold. Fixing γ^l therefore provides a stable and effective conditioning mechanism without the risk of prematurely amplifying or suppressing latent channels.

3.3 Training Objective

We supervise on mixtures with prompts spanning *speech*, *music*, and diverse (possibly compositional) *SFX*. Besides per-source reconstruction, we encourage mixture consistency by decoding the summed latent estimates $\tilde{x} = g(\sum_s \tilde{Z}_s)$. The loss maximizes SI-SDR Luo & Mesgarani (2019); Le Roux et al. (2019) for both sources and mixture:

$$\mathcal{L} = -\sum_s \text{SI-SDR}(y_s, \tilde{y}_s) \; - \; \text{SI-SDR}(x, \tilde{x}). \tag{6}$$

During training, DAC and the CLAP text encoder are frozen; we update only the FiLM-conditioned masker $Mask(\cdot)$ and the query network $query(\cdot)$.

For training and analysis, we operate on *continuous* latents $Z = Enc(x) \in \mathbb{R}^{d \times T}$: (i) gradients flow cleanly through $Mask(\cdot, e_\tau)$ and $Dec(\cdot)$ with a frozen codec (no straight-through estimators), yielding stable convergence; (ii) RVQ pretraining *regularizes Z* so pitch, timbre, onsets/transients, and textures are hierarchically organized, providing a richer, more disentangled signal for FiLM; and (iii) Z avoids run-to-run variance from codebook utilization (e.g., late RVQ sensitivity, bitrate truncation), reducing the need for special regularizers.

3.4 Deployment Discussion

For deployments with compressed bitstreams, we reconstruct embeddings by codebook lookup and use the same masker:

$$A = [a_t \in [1024]^{N_q} \mid t \in [T]], \qquad e_t \; = \; \sum_{i=1}^{N_q} \text{lookup}(a_t^{(i)}), \tag{7}$$

$$E = [e_t]_{t=1}^T \; \approx \; Z, \qquad \tilde{E}_s \; = \; M_s \odot E, \qquad \tilde{y}_s(t) \; = \; Dec(\tilde{E}_s). \tag{8}$$

When a codes-out interface is desired, we re-quantize masked embeddings and optionally decode:

$$\hat{A}_s = Quant(\tilde{E}_s), \qquad \hat{E}_s = \text{lookup}(\hat{A}_s), \qquad \tilde{y}_s(t) = Dec(\hat{E}_s). \tag{9}$$

By design, $E \approx Z$ at the operating bitrate and $Dec(E)$ already yields high-fidelity reconstructions; because our separator is a masker (selective modulation) rather than a generator, swapping $Z \rightarrow E$ preserves the semantics needed for separation with no architectural change. While we report results on Z to isolate separator performance and maintain stable optimization, we also evaluate the bitstream path by feeding reconstructed embeddings E (codes-in) to the same trained masker without any fine-tuning; performance remains competitive relative to the Z path. The residual gap can be narrowed with light fine-tuning the masker on E or optimizing an embedding-consistency loss (cf. CodecFormer-EL) in place of, or alongside, SI-SDR:

$$\mathcal{L}_{\text{emb}} = \sum_s \|\tilde{E}_s - Z_s\|_1. \tag{10}$$

In deployment, the variant simply replaces the masker input with E and optionally re-quantizes for codes-out as,

$$x(t) \xrightarrow[\text{On Edge}]{Quant(Enc(\cdot))} A \xrightarrow[\text{Codes In}]{lookup(A)} E \approx Z \xrightarrow[\text{On Server}]{Mask(E,e_\tau)} \tilde{E}_s = M_s \odot E \xrightarrow[\text{Codes Out}]{Quant(\tilde{E}_s)} \hat{A}_s. \tag{11}$$

In realistic pipelines, edge devices already run a codec and transmit code streams rather than raw audio. Traditional spectrogram-based $Spec(.)$ and waveform-based $Wave(.)$ separators, however, operate on the audio stream: they first convert audio to STFT or STFT-like representations (often via 1D convolutions), and then must *decode → separate on* X → *re-encode*, incurring additional latency and energy cost. In contrast, CodecSep performs *masking directly in the codec domain* and can output code streams without any decode–re-encode cycle.

Concretely, with codec costs C_{Enc}, C_{Dec}, spectrogram or audio-stream separator (AudioSep) cost C_{Spec}, and CodecSep masker cost C_{Mask}:

$$\text{Compute Cost for Code-stream input:} \quad \text{AudioSep} = C_{Dec} + C_{Spec} + C_{Enc}, \quad \text{CodecSep} = C_{Mask}. \tag{12}$$

We treat the codebook lookup C_{lookup} and quantization C_{Quant} costs as negligible (≈ 0) and omit the CLAP text-encoder cost since it is shared across all models. Figure 2 illustrates a typical edge–server deployment and compares compute requirements for conventional audio-stream separators (audio in → codes out) versus CodecSep's code-stream separator (codes in → codes out). As shown, the CodecSep masker operates on Z/E with small (d, T) where $|Z| \ll |X|$, dramatically reducing attention and MLP activations and enabling tighter batching and lower memory bandwidth. Interface compatibility is immediate when only codes A are available: perform a lookup to obtain E, apply the FiLM-conditioned masker, and optionally re-quantize to produce $\hat{A}_s$. CodecSep thus eliminates redundant decode/re-encode cycles in server workflows yielding low-latency, high-fidelity separation at scale. See Appendix B for a full discussion covering all of the aforementioned rationale in §3.

4 Experiments

Datasets. We evaluate across a controlled multi-stem corpus and multiple open-domain benchmarks. For in-domain experiments, we adapt Divide and Remaster v2 (dnr-v2) Petermann et al. (2022) from fixed-label three-stem separation to universal, prompt-driven separation by replacing source labels with natural-language queries. Speech and music are queried with broad category prompts (e.g., "speech", "music"), while SFX stems are queried using long-form, compositional prompts (≥ 2 overlapping sources) synthesized from FSD50K's hierarchical annotations Fonseca et al. (2022), combining fine-grained classes with parent categories (e.g., "dog barking, *Animal*, engine rumbling, *motor vehicle*"). To assess cross-domain generalization, we form three-source mixtures on AudioCaps Kim et al. (2019) (used for both training and testing in our open-domain setting) and construct test-only three-source mixtures from ESC-50 Piczak, Clotho-v2 Drossos et al. (2020), AudioSet-eval Gemmeke et al. (2017), and VGGSound Chen et al. (2020a). Dataset construction details, clip durations, split statistics, and segmentation rules are provided in the Appendix C. For dnr-v2, we report per-source scores for speech, music, and SFX. For the additional open-domain benchmarks, however, evaluation is event-conditioned rather than stem-conditioned, and many labels do not map cleanly onto the speech/music/SFX taxonomy. We therefore report dataset-level averages there, which better reflect the open-domain extraction setting; a finer semantic grouping could be explored in future analysis.

Figure 2: Typical edge–server deployment comparing compute requirements of conventional audio-stream separators (audio in → codes out) versus CodecSep discrete inference (codes in → codes out).

Evaluation. We compare CodecSep against representative spectrogram-, waveform- and codec-domain baselines—TDANet Li et al. (2023); Pons et al. (2024), Sudo rm-rf Tzinis et al. (2022a), CodecFormer Yip et al. (2024b), SDCodec Bie et al. (2024), and the text-guided audio stream separators AudioSep Liu et al. (2024), BiModalSS Mahmud et al. (2024) and Sudo rm-rf + FiLM Tzinis et al. (2022b; 2023). We report objective signal fidelity via scale-invariant signal-to-distortion ratio (SI-SDR) Luo & Mesgarani (2019); Le Roux et al. (2019) and perceptual quality via ViSQOL Chinen et al. (2020), which measures spectro–temporal similarity between the estimate $\tilde{x}$ and reference x and maps it to a 1–5 MOS-LQO scale. Following prior work (e.g., SDCodec Bie et al. (2024)), we use ViSQOL as a proxy MOS score for perceptual listening quality and complement it with a human MOS-LQS study comparing real-world outputs from CodecSep (trained on dnr-v2) and the publicly released AudioSep. To quantify efficiency, we report multiply–accumulate operations (MACs), inference time, memory footprint using `torchinfo`[1] under matched input durations (2 s) and batching (batch size 2) on a 24GB NVIDIA A30 GPU. Details on evaluation workflow for each benchmark are deferred to Appendix D.

Training. Unless otherwise stated, the DAC codec Kumar et al. (2023) and CLAP text encoder Wu et al. (2023) remain frozen; we train the FiLM-conditioned transformer masker and the lightweight query network end-to-end with an Adam optimizer Kingma & Ba (2017) and a plateau-based learning-rate schedule Mukherjee et al. (2019). We produce two variants of CodecSep, trained separately on dnr-v2 and on AudioCaps, to study distributional effects; we denote them with the suffixes "+dnr-v2" and "+AudioCaps". For fair comparison, the 3-stem versions of TDANet, Sudo rm-rf, and CodecFormer are re-trained from scratch on dnr-v2 using our setup. AudioSep is evaluated both as the publicly released checkpoint and when re-trained under matched protocols. We similarly re-train Sudo rm-rf+FiLM under the same matched settings. Pretrained checkpoints of BiModalSS and SDCodec are used as released by the authors. To reflect realistic deployments where signals traverse compression pipelines, inputs to non-codec baselines are passed through a full-band stereo-capable 48 kHz EnCodec during both training and inference. Full hyper-parameters, iteration schedules, batch configurations, and hardware details are deferred to the Appendix E.

4.1 Results and Discussions

Tables 1–8 present universal sound separation results, prompt–granularity analyses, architectural ablations, cross-dataset generalization, paraphrase robustness, and full inference complexity under matched training/evaluation protocols. Table 1 reports dnr-v2 test results for speech, music, and SFX: text-guided models

---
[1] https://github.com/tyleryep/torchinfo

Table 1: Results: Separation performance on universal sound separation (**dnr-v2-test**). To reflect the compressed-input deployment scenario studied in this work, zero-shot baselines are evaluated on *codec-processed* audio, while retrained baselines are both trained and inferred on *codec-processed* audio.

Model	Metric (↑)	Music	Speech	Sfx
AudioSep	SI-SDR	$-2.5^{\pm 4.06}$	$4.9^{\pm 4.21}$	$-0.3^{\pm 5.39}$
(zero-shot)	ViSQOL	$2.9^{\pm 0.63}$	$3.1^{\pm 0.56}$	$\mathbf{2.6^{\pm 0.77}}$
BiModalSS	SI-SDR	$-6.8^{\pm 2.73}$	$1.8^{\pm 2.78}$	$-6.36^{\pm 3.57}$
(zero-shot)	ViSQOL	$2.5^{\pm 0.57}$	$2.6^{\pm 0.51}$	$2.3^{\pm 0.72}$
AudioSep + dnr-v2	SI-SDR	$-5.6^{\pm 2.89}$	$7.7^{\pm 3.0}$	$-4.7^{\pm 3.68}$
	ViSQOL	$2.6^{\pm 0.57}$	$2.5^{\pm 0.37}$	$2.3^{\pm 0.7}$
Sudo rm-rf + FiLM + dnr-v2	SI-SDR	$-6.7^{\pm 2.62}$	$2.0^{\pm 2.76}$	$-6.6^{\pm 3.71}$
	ViSQOL	$2.7^{\pm 0.59}$	$2.9^{\pm 0.45}$	$2.3^{\pm 0.72}$
CodecSep + dnr-v2	SI-SDR	$\mathbf{1.2^{\pm 3.29}}$	$\mathbf{10.0^{\pm 2.92}}$	$\mathbf{0.9^{\pm 4.22}}$
	ViSQOL	$\mathbf{2.9^{\pm 0.57}}$	$\mathbf{3.2^{\pm 0.45}}$	$2.3^{\pm 0.73}$
CodecSep + dnr-v2	SI-SDR	$-0.2^{\pm 3.55}$	$8.3^{\pm 2.60}$	$-1.0^{\pm 4.20}$
(**codes in : codes out, zero-shot**)	ViSQOL	$2.5^{\pm 0.52}$	$3.0^{\pm 0.44}$	$2.3^{\pm 0.67}$

use generic prompts for speech/music and ground-truth compositional captions for SFX; we also include a masker ablation to isolate the role of the Transformer masker. Table 5 studies SFX prompt granularity across three regimes—(i) fixed-stem, non-text baselines (TDANet, CodecFormer, SDCodec), (ii) generic 3-stem prompts ({music, speech, sfx}), and (iii) a universal setup with fine-grained, compositional SFX prompts—thereby aligning input conditions for fair comparison with fixed-head systems. Table 6 isolates architecture: decoder-style generation (CodecFormer) vs. an unguided 3-stem masker variant and its text-guided counterpart, all operating in the codec latent space. To assess out-of-domain generalization, Table 2 benchmarks on five additional open-domain corpora (ESC-50, Clotho-v2, AudioSet, VGGSound, AudioCaps) with mixtures of three randomly sampled sources and prompts drawn from captions (not tied to fixed labels).. Finally, Table 9 compares end-to-end and architecture-only GMACs for spectrogram-domain separation versus codec-latent masking, including the practical code-stream case. All models are evaluated against the original (uncompressed) ground truth; our methods are highlighted in **bold**; and we report mean and standard deviation (1σ).

4.1.1 Source Separation Performance: In-Domain, Cross-Domain, and Subjective Evaluation

Does codec-latent masking outperform spectrogram-domain text-guided separation under matched training on dnr-v2 (cf. Table 1)? The answer is yes. CodecSep+dnr-v2 outperforms both pretrained AudioSep (zero-shot) and retrained AudioSep+dnr-v2 across all three source categories, with sizable SI-SDR gains in speech (10.0 vs. 4.9/7.7 dB), music (1.2 vs. $-2.5/-5.6$ dB), and SFX (0.9 vs. $-0.3/-4.7$ dB). In ViSQOL, CodecSep matches or exceeds AudioSep in speech and music while slightly trailing on SFX, likely reflecting differences in SFX prompt distributions (AudioSep's diverse training versus CodecSep's compositional SFX prompts derived from dnr-v2). These results indicate that masking in the structured codec latent space is more effective than spectrogram-domain text-guided separation under matched training conditions.

Does the code-stream variant remain competitive without fine-tuning (cf. Table 1)? Again, the answer is largely yes. Our *bitstream-native* variant—CodecSep+dnr-v2 (codes in: codes out, zero-shot)—evaluates the same trained masker directly on reconstructed embeddings E from code streams (§3.4) *without* any fine-tuning. Relative to the continuous-latent path, this incurs only a modest drop (roughly 1–2 dB SI-SDR across sources, with small ViSQOL deltas for music/speech and parity on SFX), yet it still surpasses AudioSep+dnr-v2 on SI-SDR for all three sources (music: -0.2 vs. -5.6 dB; speech: 8.3 vs.

Table 2: Results: Benchmarking on **ESC-50**, **Clotho-v2**, **AudioSet**, **VGGSound**, **AudioCaps**. To reflect the compressed-input deployment scenario studied in this work, the retrained baseline is both trained and inferred on *codec-processed* audio. Reported significance statistics compare **CodecSep+dnr-v2** against **AudioSep+dnr-v2** using paired tests over evaluation examples.

Model	Metric (↑)	ESC-50	Clotho-v2	AudioSet	VGGSound	AudioCaps
AudioSep	SI-SDR	$-7.8^{\pm 14.46}$	$-8.6^{\pm 17.0}$	$-7.6^{\pm 11.42}$	$-7.0^{\pm 12.65}$	$-6.4^{\pm 11.48}$
+ dnr-v2	ViSQOL	$2.3^{\pm 1.12}$	$2.1^{\pm 1.08}$	$2.1^{\pm 1.00}$	$2.2^{\pm 1.10}$	$\mathbf{2.3^{\pm 1.08}}$
CodecSep	SI-SDR	$\mathbf{-5.9^{\pm 11.55}}$	$\mathbf{-6.0^{\pm 11.10}}$	$\mathbf{-6.4^{\pm 10.53}}$	$\mathbf{-6.1^{\pm 12.12}}$	$\mathbf{-6.1^{\pm 11.62}}$
+ dnr-v2	ViSQOL	$\mathbf{2.3^{\pm 1.13}}$	$\mathbf{2.3^{\pm 1.09}}$	$\mathbf{2.2^{\pm 1.0}}$	$\mathbf{2.3^{\pm 1.11}}$	$2.2^{\pm 1.16}$
Statistical Significance						
Mean gain	SI-SDR	$+1.88$	$+2.4$	$+1.25$	$+0.92$	$+0.30$
	ViSQOL	$+0.03$	$+0.22$	$+0.04$	$+0.02$	-0.02
95% CI of gain	SI-SDR	$[1.61, 2.14]$	$[2.08, 2.72]$	$[1.11, 1.39]$	$[0.78, 1.06]$	$[0.23, 0.46]$
	ViSQOL	$[0.02, 0.04]$	$[0.18, 0.28]$	$[0.03, 0.05]$	$[0.01, 0.03]$	$[-0.03, -0.01]$
Paired t-test p-value	SI-SDR	2.77×10^{-43}	9.41×10^{-48}	8.04×10^{-66}	7.5×10^{-39}	9.72×10^{-13}
	ViSQOL	8.26×10^{-7}	4.36×10^{-5}	1.08×10^{-11}	3.23×10^{-17}	1.21×10^{-4}
Wilcoxon p-value	SI-SDR	1.18×10^{-106}	4.29×10^{-52}	5.35×10^{-67}	2.1×10^{-88}	3.23×10^{-17}
	ViSQOL	5.19×10^{-21}	1.44×10^{-8}	4.46×10^{-12}	1.61×10^{-71}	3.44×10^{-6}

7.7 dB; SFX: -1.0 vs. -4.7 dB). Compared to pretrained AudioSep (zero-shot), the codes-in:codes-out variant improves SI-SDR on speech and music, although it lags on SFX SI-SDR and ViSQOL. This shows that a deployment-friendly, *no-finetuning* bitstream path is already competitive; as discussed in §3.4, the remaining gap could be reduced with light fine-tuning on E or an embedding-consistency loss.

How does CodecSep compare with other text-guided baselines on dnr-v2 (cf. Table 1)? Beyond AudioSep, CodecSep also outperforms the USS-pretrained BiModalSS model and the retrained text-conditioned Sudo rm-rf + FiLM baseline. The heavier attention-based conditioning used in BiModalSS does not generalize well to the open-domain mixtures in dnr-v2, leading to degraded performance. CodecSep also outperforms Sudo rm-rf + FiLM under the same universal setting, likely for two reasons. First, continuous CLAP embeddings provide much richer semantic conditioning than the fixed one-hot or multi-hot vectors that Sudo rm-rf + FiLM was originally designed for, making open-domain prompting more difficult for that architecture. Second, applying FiLM across all U-Conv blocks while relying on only a single Conv1d layer for audio encoding can destabilize internal representations when conditioned with high-dimensional continuous embeddings. Since AudioSep still outperforms both of these baselines by a substantial margin, we use AudioSep as the primary baseline in the remaining experiments.

Further benchmarking on ESC-50, Clotho-v2, AudioSet, VGGSound, & AudioCaps (cf. Table 2). Extending beyond dnr-v2, we evaluate both systems on five additional open-domain benchmarks spanning environmental sounds (ESC-50), audio-captioning-style corpora (Clotho-v2, AudioCaps), weakly labeled web-scale audio (AudioSet), and visually grounded audio (VGGSound). Under matched training data and prompting protocols, CodecSep+dnr-v2 yields consistently stronger separation performance than AudioSep+dnr-v2 across all five datasets in SI-SDR, with mean gains of +1.88 dB on ESC-50, +2.4 dB on Clotho-v2, +1.25 dB on AudioSet, +0.92 dB on VGGSound, and +0.30 dB on AudioCaps. The corresponding 95% confidence intervals for SI-SDR exclude zero in every case, indicating that the observed improvements are consistently positive rather than arising from a small number of favorable examples.

A similar trend is observed in ViSQOL on four of the five benchmarks, where CodecSep attains positive mean gains of +0.03, +0.22, +0.04, and +0.02 on ESC-50, Clotho-v2, AudioSet, and VGGSound, respectively. These gains are again supported by confidence intervals that remain strictly above zero. On AudioCaps, by

Table 3: Results on *unprocessed audio*, provided for completeness. Table Table 3a presents separation performance on universal sound separation (**dnr-v2-test**). The zero-shot baselines are evaluated on raw audio in their native setting, whereas the retrained baselines are both trained and evaluated on raw audio. Table 3b presents further raw-audio benchmarking results on **ESC-50**, **Clotho-v2**, **AudioSet**, **VGGSound**, and **AudioCaps** for AudioSep retrained on **dnr-v2**.

(a) Separation performance on universal sound separation (**dnr-v2-test**) with unprocessed audio. Zero-shot baselines are evaluated in their native raw-audio setting, whereas retrained baselines are both trained and evaluated on raw audio.

Model	Metric (↑)	Music	Speech	Sfx
AudioSep	SI-SDR	$-1.1^{\pm 3.72}$	$5.4^{\pm 3.46}$	$0.5^{\pm 4.41}$
(zero-shot)	ViSQOL	$3.1^{\pm 0.52}$	$3.2^{\pm 0.46}$	$2.7^{\pm 0.63}$
BiModalSS	SI-SDR	$-6.0^{\pm 2.48}$	$2.2^{\pm 2.31}$	$-5.45^{\pm 3.21}$
(zero-shot)	ViSQOL	$2.5^{\pm 0.49}$	$2.7^{\pm 0.43}$	$2.3^{\pm 0.66}$
AudioSep + dnr-v2	SI-SDR	$-4.9^{\pm 2.61}$	$7.9^{\pm 2.52}$	$-3.9^{\pm 3.08}$
	ViSQOL	$2.7^{\pm 0.47}$	$2.6^{\pm 0.35}$	$2.3^{\pm 0.61}$
Sudo rm-rf + FiLM + dnr-v2	SI-SDR	$-5.9^{\pm 2.36}$	$3.1^{\pm 2.18}$	$-5.7^{\pm 3.27}$
	ViSQOL	$2.8^{\pm 0.50}$	$3.0^{\pm 0.39}$	$2.3^{\pm 0.65}$

(b) Further raw-audio benchmarking of AudioSep retrained on **dnr-v2** across **ESC-50**, **Clotho-v2**, **AudioSet**, **VGGSound**, and **AudioCaps**.

Model	Metric (↑)	ESC-50	Clotho-v2	AudioSet	VGGSound	AudioCaps
AudioSep	SI-SDR	$-7.0^{\pm 14.31}$	$-8.1^{\pm 16.72}$	$-7.2^{\pm 11.09}$	$-6.4^{\pm 12.48}$	$-5.9^{\pm 11.21}$
+ dnr-v2	ViSQOL	$2.3^{\pm 1.09}$	$2.2^{\pm 1.05}$	$2.1^{\pm 0.98}$	$2.3^{\pm 1.07}$	$2.3^{\pm 1.03}$

contrast, the ViSQOL difference is marginal and slightly favors AudioSep (-0.02), indicating that the two systems are broadly comparable on this perceptual metric for that benchmark even though CodecSep remains competitive in SI-SDR.

The paired statistical analysis indicates that the observed improvements are consistent across evaluation examples. For SI-SDR, the paired mean gains are positive on all five benchmarks, and the corresponding 95% confidence intervals exclude zero throughout. Both the paired *t*-test and the Wilcoxon signed-rank test likewise indicate statistically significant differences in favor of CodecSep across all datasets. A similar pattern is observed for ViSQOL on ESC-50, Clotho-v2, AudioSet, and VGGSound, where positive paired gains are again supported by confidence intervals above zero and significant paired tests. On AudioCaps, the ViSQOL difference is small and slightly favors AudioSep, indicating broadly comparable perceptual quality on that benchmark despite CodecSep's competitive separation fidelity. Overall, these results suggest that the gains of CodecSep are not confined to the in-domain dnr-v2 setting, but generalize consistently across a diverse set of open-domain benchmarks.

Why evaluate non-codec baselines on codec-processed audio? We route spectrogram- and waveform-domain baselines through codec processing to compare methods in the deployment regime that motivates CodecSep. In realistic edge–server workflows, audio is typically available as codec-compressed bitstreams rather than as ideal raw waveforms. Under such conditions, non-codec separators must first decode the signal, perform separation in the audio or spectrogram domain, and then re-encode if codec-compatible output is required. CodecSep is explicitly designed to avoid this decode–separate–re-encode overhead by operating directly on codec representations. Accordingly, training and evaluating non-codec baselines on codec-processed audio reflects the compressed-input conditions under which these systems would actually be deployed. We stress, however, that this setup is intended to assess *deployment realism* and interface compatibility, rather than to represent the native best-case raw-audio performance of the non-codec baselines.

Native raw-audio performance of non-codec baselines (cf. Table 3). For completeness, we also report the performance of the non-codec baselines in their native *raw-audio* setting, without codec processing (cf. Table 3). These results help separate two effects: the intrinsic separation capability of the baseline architectures in their preferred operating regime, and their performance under the codec-mediated deployment setting that motivates CodecSep. On **dnr-v2-test** (cf. Table 3a), evaluating AudioSep on raw audio improves its absolute scores relative to the codec-processed setting, as expected, since the model is no longer exposed to codec-induced mismatch. A similar trend is observed for the retrained AudioSep+dnr-v2 and the other non-codec baselines. Even so, CodecSep+dnr-v2 evaluated in the codec-processed regime (Table 1) still exceeds the raw-audio AudioSep baselines in SI-SDR on all three dnr-v2 stems, while remaining broadly competitive in perceptual quality, though the raw-audio AudioSep zero-shot model retains an advantage in ViSQOL on music and SFX. Likewise, on the additional open-domain benchmarks (cf. Table 3b), AudioSep+dnr-v2 evaluated on unprocessed audio yields modestly stronger absolute scores than its codec-processed counterpart. However, CodecSep+dnr-v2 in the codec-processed setting (Table 2) still maintains consistently stronger SI-SDR across all five datasets, with broadly comparable ViSQOL and only a small deficit on AudioCaps. These raw-audio results clarify that codec processing does affect non-codec separators in absolute terms. At the same time, they do not alter the main claim of this work: CodecSep is designed specifically for the *codec-mediated deployment* regime, where audio is exchanged as compressed bitstreams. Our primary comparisons therefore remain those in Tables 1 and 2, which evaluate all methods under the compressed-input conditions relevant to that deployment setting.

Subjective evaluation (MOS–LQS) (cf. Table 4). We ran a human evaluation test with $n=20$ participants on 20 dnr-v2 3-stem test mixtures, comparing paired outputs from CodecSep+dnr-v2 and the official AudioSep model using fixed *speech/music* prompts and per-clip *sfx* prompts. We used the official pretrained *AudioSep* model, rather than our retrained variant, because the pretrained checkpoint performed substantially better on *dnr-v2*; notably, this pretrained model was trained on approximately 14,100 hours of audio from multiple datasets with diverse natural-language prompts. Raters scored each stem independently in randomized order on the MOS–LQS scale (1=bad, 5=excellent); we report mean $\pm 1\sigma$. As shown in Table 4, CodecSep scored $3.34^{\pm1.00}$ vs. AudioSep $2.61^{\pm1.04}$ overall. By source, CodecSep achieved $3.17^{\pm1.01}$ (music), $3.37^{\pm0.97}$ (sfx), and $3.49^{\pm1.00}$ (speech), while AudioSep obtained $2.49^{\pm0.95}$, $2.84^{\pm1.16}$, and $2.50^{\pm1.02}$, respectively. These outcomes align with objective trends (SI-SDR/ViSQOL) and indicate consistent perceptual gains for CodecSep.

We further assessed statistical significance using paired tests on the *per-clip* MOS–LQS means over 20 test clips. CodecSep significantly outperformed AudioSep on all stems as well as on the overall score. For the overall MOS–LQS, CodecSep improved the clip-level mean from 2.61 to 3.34, yielding a mean paired gain of 0.74 points (95% CI: [0.55, 0.92], paired t-test: $p = 9.78 \times 10^{-8}$; Wilcoxon signed-rank: $p = 1.03 \times 10^{-4}$). The improvements were also significant for Music (mean gain: 0.68, 95% CI: [0.47, 0.91], $p = 2.74 \times 10^{-6}$), Speech (mean gain: 0.99, 95% CI: [0.78, 1.20], $p = 6.55 \times 10^{-9}$), and SFX (mean gain: 0.53, 95% CI: [0.20, 0.84], $p = 2.81 \times 10^{-3}$). These findings indicate that the subjective preference for CodecSep is consistent across clips and not driven by a small subset of examples. Paired model outputs and reference stems are included in the supplementary materials for side-by-side listening.

4.1.2 Effect of Prompt Granularity on Source Separation Performance

How does SFX prompt granularity affect separation quality, and why include fixed-stem baselines here (cf. Tables 1–5)? We use this experiment primarily as an *analysis of prompt granularity*, rather than as a direct competitor comparison. To this end, we evaluate three regimes: (i) *fixed-stem* baselines without text guidance (TDANet, Sudo rm-rf, CodecFormer, SDCodec) (cf. Table 5), included only to anchor performance under closed-set, fixed-output separation; (ii) *generic 3-stem* prompting using {"music", "speech", "sfx"} (cf. Table 5); and (iii) *universal* prompting that retains generic cues for speech/music but replaces the SFX label with fine-grained, compositional SFX descriptions (cf. Table 1). For (ii) and (iii), we train and evaluate separate versions of both CodecSep and AudioSep under the corresponding prompt setting.

Does CodecSep remain effective under coarse generic prompts (cf. Table 5)? Yes. Under matched generic-prompt training and evaluation, CodecSep remains robust and continues to surpass spectrogram-

Table 4: Subjective evaluation (MOS–LQS) on **dnr-v2** 3-stem mixtures. Mean $\pm$ standard deviation across n=20 raters and 20 test clips. Each stem was rated independently (1=bad, 5=excellent). Statistical significance is computed using paired tests on *per-clip* mean ratings (n=20 clips), comparing CodecSep against AudioSep.

Model	Overall ($\uparrow$)	Music ($\uparrow$)	Speech ($\uparrow$)	Sfx ($\uparrow$)
AudioSep	2.61 ± 1.04	2.49 ± 0.95	2.50 ± 1.02	2.84 ± 1.16
CodecSep + dnr-v2	$\mathbf{3.34 \pm 1.00}$	$\mathbf{3.17 \pm 1.01}$	$\mathbf{3.49 \pm 1.00}$	$\mathbf{3.37 \pm 0.97}$
Statistical Significance				
Mean gain	$+0.74$	$+0.68$	$+0.99$	$+0.53$
95% CI of gain	$[0.55, 0.92]$	$[0.47, 0.91]$	$[0.78, 1.20]$	$[0.20, 0.84]$
Paired t-test p-value	9.78×10^{-8}	2.74×10^{-6}	6.55×10^{-9}	2.81×10^{-3}
Wilcoxon p-value	1.03×10^{-4}	2.29×10^{-4}	8.82×10^{-5}	8.48×10^{-3}

Table 5: Results: Impact of SFX Prompt Granularity on Universal Sound Separation (**dnr-v2-test**)

Model	Metric ($\uparrow$)	Music	Speech	Sfx
3-Stem: Fixed stem baselines (no text-guidance)				
TDANet	SI-SDR	$1.8^{\pm3.55}$	$10.2^{\pm2.91}$	$1.4^{\pm4.90}$
	ViSQOL	$2.9^{\pm0.58}$	$3.1^{\pm0.43}$	$2.4^{\pm0.72}$
Sudo rm-rf	SI-SDR	$-0.9^{\pm4.01}$	$9.0^{\pm2.60}$	$0.6^{\pm4.74}$
	ViSQOL	$2.7^{\pm0.59}$	$2.9^{\pm0.45}$	$2.3^{\pm0.72}$
CodecFormer	SI-SDR	$-5.7^{\pm3.44}$	$2.3^{\pm2.32}$	$-6.5^{\pm4.36}$
	ViSQOL	$2.2^{\pm0.47}$	$2.5^{\pm0.49}$	$2.1^{\pm0.67}$
SDCodec	SI-SDR	$\mathbf{1.9^{\pm3.68}}$	$\mathbf{11.3^{\pm2.98}}$	$\mathbf{1.8^{\pm4.08}}$
	ViSQOL	$\mathbf{3.0^{\pm0.56}}$	$\mathbf{3.5^{\pm0.40}}$	$\mathbf{2.6^{\pm0.73}}$
3-Stem: {"music","speech","sfx"} as generic prompt				
AudioSep	SI-SDR	$\mathbf{-2.5^{\pm4.06}}$	$4.9^{\pm4.21}$	$-6.7^{\pm4.73}$
(zero-shot)	ViSQOL	$\mathbf{2.9^{\pm0.63}}$	$\mathbf{3.1^{\pm0.56}}$	$2.1^{\pm0.68}$
AudioSep + dnr-v2	SI-SDR	$-6.2^{\pm2.77}$	$\mathbf{7.7^{\pm3.11}}$	$-2.1^{\pm3.90}$
	ViSQOL	$2.6^{\pm0.57}$	$2.5^{\pm0.37}$	$2.4^{\pm0.74}$
CodecSep + dnr-v2	SI-SDR	$-7.7^{\pm2.84}$	$4.6^{\pm2.48}$	$\mathbf{0.6^{\pm4.15}}$
	ViSQOL	$2.5^{\pm0.55}$	$2.7^{\pm0.49}$	$\mathbf{2.4^{\pm0.70}}$

domain AudioSep, indicating that its effectiveness does not rely on unusually careful prompt engineering. Relative to the non-text-guided fixed-stem systems, CodecSep also remains competitive on SFX while maintaining comparable speech and music quality. In particular, its USS variant outperforms Sudo rm-rf across all stems with modest margins (cf. Table 5).

What is gained by replacing the generic "sfx" label with fine-grained SFX descriptions (cf. Tables 1–5)? The main gain is improved scene-level disentanglement. Replacing the generic "sfx" prompt with detailed, compositional SFX descriptions consistently sharpens SFX extraction and, importantly, also improves speech and music stems. This suggests that richer SFX supervision does not merely help the queried SFX stem; rather, it helps the model partition the mixture more cleanly overall. The fixed-stem baselines are therefore useful here only as a closed-set reference point: the key finding is not that CodecSep must beat

Table 6: Results: Architectural advantages in using CodecFormer decoder as masker (**dnr-v2-test**)

Model	Metric (↑)	Music	Speech	Sfx
CodecFormer	SI-SDR	$-5.8^{\pm 3.44}$	$2.3^{\pm 2.32}$	$-6.5^{\pm 4.36}$
	ViSQOL	$2.2^{\pm 0.47}$	$2.5^{\pm 0.49}$	$2.1^{\pm 0.67}$
CodecSep + dnr-v2	SI-SDR	$1.2^{\pm 3.35}$	$10.0^{\pm 2.91}$	$0.9^{\pm 4.18}$
(unguided, 3-stem)	ViSQOL	$2.8^{\pm 0.55}$	$3.1^{\pm 0.45}$	$\mathbf{2.5^{\pm 0.72}}$
CodecSep + dnr-v2	SI-SDR	$\mathbf{1.2^{\pm 3.29}}$	$\mathbf{10.0^{\pm 2.92}}$	$\mathbf{0.9^{\pm 4.22}}$
(text-guided)	ViSQOL	$\mathbf{2.9^{\pm 0.57}}$	$\mathbf{3.2^{\pm 0.45}}$	$2.3^{\pm 0.73}$
CodecSep + dnr-v2	SI-SDR	$-6.8^{\pm 2.77}$	$2.0^{\pm 2.84}$	$-6.8^{\pm 3.83}$
(ablate **Masker**)	ViSQOL	$2.5^{\pm 0.58}$	$2.6^{\pm 0.50}$	$2.1^{\pm 0.74}$

them in every case, but that finer-grained semantic supervision improves universal prompt-driven separation beyond what generic three-stem prompting provides.

These controlled studies cover multiple prompt granularities, but they also suggest a broader direction: training on larger and more diverse corpora with a wider spectrum of prompt specificities may yield further gains, which we leave for future work.

4.1.3 Architectural Choice in Codec Latent Space: Masking vs. Generation

Is the Transformer masker itself necessary (cf. Table 6)? The ablation suggests that it is. The lightweight CodecSep+dnr-v2 (ablate Masker) removes the transformer masker and applies FiLM directly to the encoder. While this variant attains SI-SDR comparable to AudioSep+dnr-v2 (cf. Table 1) and yields better perceptual speech quality, its overall separation quality drops relative to full CodecSep. This supports our architectural choice: FiLM modulation is more effective when applied in a dedicated masker than when injected directly into the encoder, where it perturbs the mixture latents themselves.

Is masking better than generation in codec latent space (cf. Table 6)? To isolate this architectural question, we compare (i) CodecFormer, which performs decoder-style source generation, (ii) CodecSep (unguided, 3-stem), which repurposes the CodecFormer Transformer as a masker over codec latents, and (iii) CodecSep (text-guided), which adds prompt conditioning on top of the same masking formulation.

What changes when decoder-style generation is replaced by masking? The results on dnr-v2-test show a clear and consistent pattern: replacing decoder-style generation with masking strengthens separation across music, speech, and SFX. This supports our design rationale that, in the DAC latent domain, it is more effective to modulate existing, semantically structured content than to synthesize new source latents from scratch. In particular, masking reduces artifacts and cross-talk leakage, preserves long-range periodicity, timbre, and transient organization already encoded by the codec, and yields a more stable optimization than end-to-end generation.

What is gained by adding text guidance on top of masking? Text conditioning provides a further uniform improvement over the unguided masker. This suggests that once separation is formulated as latent selection, semantic prompting can steer that selection more precisely. Put differently, the masker formulation concentrates Transformer capacity on deciding *where* and *how much* information to pass, rather than *what* new content to generate. This is exactly the operating regime we want in structured codec latent space.

4.1.4 Qualitative Evidence for Structured Source Organization in Codec Latent Space

Does the codec latent space exhibit source-dependent structure? Figure. 3 provides direct qualitative evidence that the frozen codec encoder organizes mixtures into a latent representation that already preserves source-discriminative structure. The reference source latents for speech (Figure. 3c), music (Figure. 3e), and SFX (Figure. 3g) display visibly different activation patterns across channels and time, rather than

(a) Mixture encoder latents Z.

(b) Absolute latent residual, $|Z - \frac{1}{3}(Z_{speech} + Z_{music} + Z_{sfx})|$.

(c) Reference speech latents Z_{speech}.

(d) Estimated speech mask M_{speech}.

(e) Reference music latents Z_{music}.

(f) Estimated music mask M_{music}.

(g) Reference SFX latents Z_{SFX}.

(h) Estimated SFX mask M_{SFX}.

Figure 3: Qualitative latent-space analysis for a representative dnr-v2 mixture (clip 1590). We visualize the mixture encoder latent Z in Fig. 3a, the corresponding reference source latents for speech Z_{speech}, music Z_{music}, and SFX Z_{SFX} in Figs. 3c, 3e, and 3g, the estimated source-conditioned masks M_{speech}, M_{music}, and M_{SFX} in Figs. 3d, 3f, and 3h, and the absolute residual $\left|Z - \frac{1}{3}(Z_{\mathrm{speech}} + Z_{\mathrm{music}} + Z_{\mathrm{SFX}})\right|$ in Fig. 3b. The reference source latents exhibit visibly different patterns across sources. By contrast, the estimated masks are predominantly channel-wise: each mask remains nearly constant along the temporal axis and varies mainly across latent channels. This suggests that CodecSep separates sources primarily through source-conditioned reweighting and attenuation of latent channels rather than strongly time-localized masking. The residual remains sparse over broad regions of the latent plane, providing qualitative support for the view that the mixture latent retains a meaningful relationship to the latent organization of its constituent sources, without implying exact disentanglement or linear compositionality. The corresponding audio clip and separated outputs are included in the supplementary material for side-by-side listening.

appearing as weak perturbations of a shared, unstructured representation. This observation supports the core hypothesis underlying CodecSep: modern neural audio codec latents are sufficiently structured for source extraction to be performed through masking, without requiring source generation or explicit re-encoding.

What do the estimated masks reveal about the separation mechanism? The most striking feature of the learned masks is that they are explicitly *channel-wise*. As seen in Figures. 3d, 3f, and 3h, each estimated mask forms nearly horizontal bands with essentially constant values over time. In other words, the masks

do not exhibit meaningful temporal variation; instead, they act by selecting and attenuating specific latent channels according to the text query. This qualitative behavior is important, because it shows that CodecSep does not separate sources by applying sharply localized temporal gating. Rather, the model operates by modulating a structured latent basis in a source-dependent manner. This also helps explain the slight residual leakage of non-target sources that can still be heard in some separated outputs in the supplementary audio examples: because the masks re-weight shared latent channels rather than enforcing perfectly source-exclusive, time-localized suppression, channels that carry partially overlapping information may remain weakly active in the reconstructed signal.

What does the residual analysis show? To further examine the structured-latent hypothesis, we compare the mixture latent with the mean of the corresponding reference source latents through the absolute residual

$$\left| Z - \frac{1}{3} \left(Z_{\text{speech}} + Z_{\text{music}} + Z_{\text{SFX}} \right) \right|. \tag{13}$$

The residual remains sparse over broad regions, suggesting that the mixture latent retains a visible relationship to the latent organization of its constituent sources, rather than appearing completely unstructured. This provides *qualitative support* for the view that the codec latent space preserves some source-related organization. At the same time, we do not interpret this as evidence of exact source disentanglement or linear compositionality, especially because DAC was not trained with an explicit source-disentanglement objective. In dnr-v2, mixtures are generated through a non-trivial mixing and normalization pipeline rather than by simple uniform averaging of the constituent source waveforms (cf. Appendix C). Consequently, the resulting mixture latent need not coincide exactly with an equal-weight combination of the source latents, and some residual activity is therefore expected.

What is the main takeaway from this analysis? Taken together, these heatmaps provide direct qualitative support for the design premise of CodecSep. The codec latent space is not merely compact; it also appears to preserve meaningful source-dependent organization, and the learned masks exploit this structure primarily through channel-wise modulation. This qualitative evidence complements the architectural results in Table 6, where masking-based separation outperforms decoder-style latent generation, and supports our interpretation that source extraction in CodecSep is enabled by structured source organization already present in the codec representation. For transparency and independent inspection, the corresponding clip and separated outputs are provided in the supplementary material for side-by-side listening.

4.1.5 Oracle Codec Reconstruction vs. Separated Output

How much of the observed distortion comes from the codec bottleneck itself (cf. Table 7)? To address this directly, we compare CodecSep's separated outputs against a codec-only oracle reference. In the oracle setting, no separator is used: each clean reference signal—the mixture waveform and each ground-truth source stem—is independently passed through the frozen DAC encoder–decoder, and the reconstructed signal is then evaluated against its original reference. This isolates the distortion introduced purely by the codec backbone and therefore provides a useful reference point for interpreting the artifacts seen in separated outputs.

What does the oracle comparison show? The DAC oracle achieves strong reconstruction quality on the mixture and reference stems, with SI-SDR / ViSQOL of 6.3/4.1 for the mixture, 6.8/3.8 for music, 7.4/4.2 for speech, and 1.8/3.8 for SFX. These values indicate that the codec itself is not lossless, and that part of the degradation observed in CodecSep outputs is inherited from the frozen DAC reconstruction bottleneck. This is especially clear in ViSQOL, where CodecSep remains below the oracle across all categories.

What artifacts are due to separation rather than the codec alone? Comparing CodecSep against the oracle shows that the remaining gap is source-dependent. For mixture reconstruction, CodecSep trails the oracle (4.1 vs. 6.3 SI-SDR; 3.7 vs. 4.1 ViSQOL), indicating that the separation process introduces additional distortion beyond codec reconstruction alone. A similar pattern holds for music and SFX, where CodecSep remains below the oracle in both SI-SDR and ViSQOL. This suggests that these stems are still limited primarily by separation error—e.g., incomplete source selection, residual interference, or masking imprecision—rather than by the codec alone.

Table 7: Oracle codec reconstruction versus separated outputs on **dnr-v2-test**. The DAC (Oracle) rows report codec-only reconstruction of the corresponding reference signals with *no separation model*: the mixture waveform and each ground-truth stem are passed independently through the frozen DAC encoder–decoder and then evaluated against their original references. This provides a codec-limited reference that isolates distortion introduced by the codec itself. The CodecSep + dnr-v2 rows report the actual separated outputs obtained by prompt-guided masking in codec latent space. Comparing the two clarifies how much degradation is attributable to the codec bottleneck versus the separation stage.

Model	Metric ($\uparrow$)	Mixture	Music	Speech	Sfx
DAC	SI-SDR	$6.3^{\pm 2.35}$	$6.8^{\pm 4.3}$	$7.4^{\pm 3.1}$	$1.8^{\pm 5.4}$
(Oracle)	ViSQOL	$4.1^{\pm 0.15}$	$3.8^{\pm 0.27}$	$4.2^{\pm 0.15}$	$3.8^{\pm 0.33}$
CodecSep + dnr-v2	SI-SDR	$4.1^{\pm 2.06}$	$1.2^{\pm 3.29}$	$\mathbf{10.0^{\pm 2.92}}$	$0.9^{\pm 4.22}$
(text-guided)	ViSQOL	$3.7^{\pm 0.22}$	$2.9^{\pm 0.57}$	$3.2^{\pm 0.45}$	$2.3^{\pm 0.73}$

At the same time, this gap should not be interpreted as arising solely from separator imperfections. In dnr-v2, the separated outputs are generated from *mixtures*, whereas the oracle source rows are obtained by reconstructing the corresponding *clean source stems* directly. Because the dnr-v2 mixtures are created through a non-trivial mixing and normalization procedure (cf. Appendix C), recovering a target stem from the mixture is inherently more challenging than direct codec reconstruction of an isolated clean source. The oracle comparison therefore isolates codec-induced distortion, but the remaining gap also reflects the intrinsic difficulty of mixture-conditioned source recovery on dnr-v2.

Why can speech SI-SDR exceed the oracle? Speech shows a different pattern: CodecSep attains higher SI-SDR than the DAC oracle (10.0 vs. 7.4), while still remaining clearly below it in ViSQOL (3.2 vs. 4.2). This is best understood as a difference between *signal-level* and *perceptual* evaluation. The DAC oracle reflects codec reconstruction fidelity of the clean speech stem and therefore captures waveform distortion introduced by the codec. By contrast, SI-SDR mainly measures target-aligned signal energy after optimal scaling and is not designed to reflect perceptual naturalness. If the separator preserves the dominant speech structure while effectively suppressing interfering mixture components, it can achieve a higher SI-SDR even though the resulting waveform still contains masking artifacts or reduced perceptual fidelity. The lower ViSQOL for CodecSep confirms exactly this point: speech separation is strong in a signal-recovery sense, but perceptual quality remains constrained by codec and masking distortions.

What is the main takeaway from this analysis? This experiment clarifies that the observed artifacts in CodecSep arise from multiple factors: the reconstruction bottleneck imposed by the frozen DAC backbone, the difficulty of recovering sources from dnr-v2's loudness-controlled mixtures, and an additional separation gap introduced by imperfect masking. The oracle comparison shows that the codec already preserves strong perceptual quality, while the remaining gap for mixture, music, and SFX indicates that a substantial portion of the error is due to the separation stage itself. For speech, the higher SI-SDR but lower ViSQOL of CodecSep relative to the oracle highlights an important metric distinction: signal-level recovery can improve even when perceptual fidelity remains bounded by codec and masking artifacts. Overall, the results show that CodecSep is not limited by codec reconstruction alone; the residual errors reflect codec-induced distortion, mixture-formation difficulty, and separation-specific imperfections.

4.1.6 Source Separation Robustness Under Prompt Paraphrasing

Does CodecSep remain effective under prompt paraphrasing (cf. Table 8)? To probe lexical sensitivity, we re-evaluate both CodecSep + dnr-v2 and AudioSep + dnr-v2 on the dnr-v2 test split by replacing the generic training-time prompts for *speech* and *music* with three unseen paraphrases per class—*speech*: {"spoken voice", "human conversation", "people talking"}; *music*: {"instrumental music", "band playing", "melody with instruments"}. This constitutes a zero-shot paraphrase generalization test: the models are trained with generic category cues but must respond to synonymic, potentially broader descriptors at inference.

Table 8: Results: Using ambiguous prompts for Speech and Music (**dnr-v2-test**)

Model	Metric (↑)	Music	Speech
AudioSep + dnr-v2	SI-SDR	$-6.4^{\pm 3.29}$	$4.1^{\pm 3.77}$
	ViSQOL	$2.5^{\pm 0.57}$	$2.6^{\pm 0.47}$
CodecSep + dnr-v2	SI-SDR	$\mathbf{-5.6^{\pm 3.61}}$	$\mathbf{4.2^{\pm 4.18}}$
	ViSQOL	$\mathbf{2.6^{\pm 0.58}}$	$\mathbf{2.7^{\pm 0.51}}$

How do CodecSep and AudioSep behave under lexical variation? Both exhibit the expected degradation when moving from generic to paraphrased prompts, confirming that lexical ambiguity weakens query–audio alignment. However, CodecSep degrades more gracefully overall, maintaining stronger separation and perceptual quality for *speech*, while also retaining a small but persistent advantage for *music*. Although the gap between the two systems narrows under paraphrasing, the relative ranking is preserved. This suggests that FiLM-conditioned masking over structured codec latents confers a degree of robustness to synonym-level prompt shifts.

What does this experiment establish, and what does it not test? This experiment isolates lexical paraphrases only. We do not consider prompts with explicit temporal or relational structure (e.g., "applause follows a song"), which would require the model to respond not just to synonymy but also to event ordering or compositional temporal cues. We leave such prompt variations for future work.

4.1.7 Efficiency in Code-Stream Deployment

What is gained in code-stream deployment (cf. Table 9)? The principal efficiency advantage of CodecSep appears in the *code-stream* setting, where separation is performed directly on codec representations rather than on decoded waveforms. This is the practically relevant regime for edge–server systems, where audio is typically transmitted, stored, and exchanged as codec bitstreams. In this setting, CodecSep is markedly more efficient than spectrogram-domain and waveform-domain separators across all measured axes. We therefore interpret the headline efficiency gains of CodecSep primarily as a *deployment-level* advantage in codec-mediated pipelines, rather than as a universal claim about raw-audio separation cost.

How large are the compute and latency gains in code-stream mode? In terms of hardware-agnostic compute (cf. Table 9a), CodecSep requires only 1.35 GMACs when operating on codec bitstreams, compared to 73.6 GMACs for AudioSep and 56.5 GMACs for Sudo rm-rf, yielding roughly 54× and 40× reductions, respectively (and 25×/12× under the architecture-only comparison). These savings translate directly to latency (cf. Table 9b): CodecSep achieves 0.19 s inference in code-stream mode—about 8× faster than AudioSep and approximately 10× faster than Sudo rm-rf.

What is gained in parameter and memory footprint? Parameter efficiency follows the same trend (cf. Table 9d): when fed code streams, CodecSep uses only 16.3M parameters, substantially smaller than AudioSep (112.8M) and Sudo rm-rf (78.8M), reflecting its lightweight masker-only design. The memory savings are even more pronounced. For forward/backward activations (cf. Table 9e), CodecSep requires only 28 MB in code-stream mode, compared to 1.58 GB for AudioSep and 2.84 GB for Sudo rm-rf—over 50× to 100× reductions. The same pattern holds for full memory footprint (cf. Table 9f): CodecSep uses only 76.5 MB, versus 2.03 GB for AudioSep and 3.15 GB for Sudo rm-rf, corresponding to roughly 27× and 41× reductions, respectively.

How should the audio-stream results be interpreted? In the *audio-stream* setting, CodecSep must additionally run the codec encoder and decoder, so its end-to-end compute is *not* lower than AudioSep. Accordingly, the audio-stream comparison should be interpreted as showing that CodecSep remains broadly comparable in this regime, rather than uniformly more efficient. This distinction is important: the main efficiency claim of CodecSep is not that it is cheaper for generic waveform-side separation, but that it becomes substantially more efficient when separation is carried out directly on code streams. Even in the audio-stream

Table 9: Full Inference Compute Benchmarking Across Six Settings on 24GB NVIDIA A-30 GPU

(a) Inference GMACs (↓)

Model	Audio Stream I/O	Code Stream I/O	Architecture-only
AudioSep	33.5	73.6	33.5
Sudo rm-rf	**16.44**	56.54	16.44
CodecSep	41.45	**1.35**	**1.35**

Codec GMACs: Enc=12.28, Dec=27.82

(b) Inference Time (s) (↓)

Model	Audio Stream I/O	Code Stream I/O	Architecture-only
AudioSep	**0.33**	1.63	0.33
Sudo rm-rf	0.51	1.81	0.51
CodecSep	1.49	**0.19**	**0.19**

Codec Inference Times (s): Enc=1.16, Dec=0.14

(c) Parameter count (in million) (↓)

Model	Audio Stream I/O	Code Stream I/O	Architecture-only
AudioSep	39	112.8	39
Sudo rm-rf	**5**	78.8	**5**
CodecSep	90.1	**16.3**	16.3

Codec Parameter Count (in million): Enc=21.5, Dec=52.3

(d) Parameter-only Memory Footprint (MB) (↓)

Model	Audio Stream I/O	Code Stream I/O	Architecture-only
AudioSep	156.13	447.62	156.13
Sudo rm-rf	**20.07**	311.25	**20.07**
CodecSep	339.89	**48.4**	48.4

Codec Parameter-only Memory Footprint (MB): Enc=85.1, Dec=206.39

(e) Forward/Backward Pass Memory Footprint (MB) (↓)

Model	Audio Stream I/O	Code Stream I/O	Architecture-only
AudioSep	**804.3**	1580.6	804.3
Sudo rm-rf	2032.13	2836.43	2032.13
CodecSep	804.4	**28.06**	**28.06**

Codec Forward/Backward Pass Memory Footprint (MB): Enc=310.48, Dec=465.82

(f) Full Memory Footprint (MB) (↓)

Model	Audio Stream I/O	Code Stream I/O	Architecture-only
AudioSep	**960.43**	2028.22	960.43
Sudo rm-rf	2052.2	3147.68	2052.2
CodecSep	1144.25	**76.46**	**76.46**

Codec Full Memory Footprint (MB): Enc=395.58., Dec=672.21

Table 10: Results: Extending CodecSep to 48 kHz full-band (**dnr-v2-test**)

Model	Sampling Rate	Metric (↑)	Music	Speech	Sfx
AudioSep	32 kHz	SI-SDR	$-2.5^{\pm4.06}$	$4.9^{\pm4.21}$	$-0.3^{\pm5.39}$
(zero-shot)		ViSQOL	$2.9^{\pm0.63}$	$3.1^{\pm0.56}$	$\mathbf{2.6^{\pm0.77}}$
AudioSep + dnr-v2	32 kHz	SI-SDR	$-5.6^{\pm2.89}$	$7.7^{\pm3.0}$	$-4.5^{\pm3.68}$
		ViSQOL	$2.6^{\pm0.57}$	$2.5^{\pm0.37}$	$2.3^{\pm0.7}$
CodecSep + dnr-v2	16 kHz	SI-SDR	$\mathbf{1.2^{\pm3.29}}$	$\mathbf{10.0^{\pm2.92}}$	$\mathbf{0.9^{\pm4.22}}$
(DAC Backbone)		ViSQOL	$\mathbf{2.9^{\pm0.57}}$	$\mathbf{3.1^{\pm0.45}}$	$2.3^{\pm0.73}$
CodecSep + dnr-v2	24 kHz	SI-SDR	$0.2^{\pm3.3}$	$8.8^{\pm2.9}$	$0.6^{\pm4.2}$
(DAC Backbone)		ViSQOL	$2.7^{\pm0.56}$	$3.0^{\pm0.44}$	$2.3^{\pm0.72}$
CodecSep + dnr-v2	44.1 kHz	SI-SDR	$-2.3^{\pm3.27}$	$5.9^{\pm2.48}$	$-0.3^{\pm3.79}$
(DAC Backbone)		ViSQOL	$2.5^{\pm0.46}$	$2.7^{\pm0.42}$	$2.4^{\pm0.68}$
CodecSep + dnr-v2	48 kHz	SI-SDR	$-2.8^{\pm3.5}$	$5.4^{\pm2.36}$	$-0.5^{\pm3.83}$
(EnCodec Backbone)		ViSQOL	$2.4^{\pm0.5}$	$2.6^{\pm0.41}$	$2.4^{\pm0.65}$

setting, however, the results remain informative: CodecSep still uses less memory than Sudo rm-rf (1.14 GB vs. 2.05 GB), indicating that codec overhead, while non-trivial, is not itself the dominant bottleneck.

Taken together, these results clarify the intended efficiency claim of CodecSep. Its main advantage is not a universal reduction in raw-audio separation cost, but a substantial *systems-level* gain in codec-mediated deployment, where operating directly on code streams avoids the decode–separate–re-encode pathway required by conventional baselines. That is precisely the regime in which compute, memory, and latency all improve dramatically, making CodecSep especially well-suited to scalable edge–server pipelines.

4.1.8 Bandwidth Scaling: Extending CodecSep to Full-Band Audio

How should the sampling-rate mismatch with AudioSep be interpreted, and does the masking formulation remain effective as bandwidth increases (cf. Table 10)? The main CodecSep system in this paper uses a **16 kHz DAC backbone**. This choice is deliberate: our primary objective is to study *low-MAC, codec-mediated separation,* and the 16 kHz DAC variant provides the most favorable tradeoff

between separation quality and deployment efficiency. By contrast, the official AudioSep checkpoint is available as a **32 kHz** model trained on a much larger and more diverse corpus (roughly 15K hours), and for fairness we also retrain the same 32 kHz AudioSep variant under our matched training protocol. We do not alter AudioSep to another sampling rate, since doing so would require nontrivial redesign and careful hyperparameter re-optimization, introducing a new confound by evaluating a materially altered version of the baseline rather than the standard model itself.

At the same time, lower bandwidth can simplify separation, so this mismatch should not simply be ignored. To contextualize it directly, we perform a dedicated *bandwidth-scaling study* for CodecSep by swapping the frozen codec backbone from 16 kHz DAC to higher-bandwidth codecs while keeping the FiLM-conditioned masker and training objective unchanged. Specifically, we evaluate **24 kHz** and **44.1 kHz** DAC variants as well as a **48 kHz EnCodec** backbone. Although our paper targets *mono* separation, the 48 kHz EnCodec experiment is still carried out in the same mono setting so that the comparison remains architectural rather than spatial.

Why does separation become harder at higher bandwidth? As the sampling rate F_s increases, the representation must account for progressively richer high-frequency structure, finer temporal transients, and more densely packed spectral detail. In codec latent space, these additional details are typically less cleanly isolated than the lower-frequency coarse structure and tend to become more tightly entangled across sources. As a result, a masker-based separator must make more delicate source-selection decisions over a denser and less separable latent organization. At the same time, the latent sequence length typically increases ($T \uparrow$), which raises both modeling difficulty and compute. Together, these effects make full-band separation intrinsically harder than narrow-band or mid-band separation, even though the underlying masking formulation remains unchanged.

What pattern is revealed by Table 10? The results show a clear trend: for a fixed masker capacity, *absolute* SI-SDR and, to a lesser extent, ViSQOL generally decrease as sampling rate increases. Relative to CodecSep at **16 kHz** (1.2/10.0/0.9 SI-SDR and 2.9/3.1/2.3 ViSQOL for music/speech/SFX), the **24 kHz** DAC variant remains strong but drops modestly (0.2/8.8/0.6 SI-SDR; 2.7/3.0/2.3 ViSQOL), while the **44.1 kHz** DAC and **48 kHz** EnCodec variants degrade further ($-2.3/5.9/-0.3$ and $-2.8/5.4/-0.5$ SI-SDR, respectively). This confirms that the 16 kHz operating point is indeed favorable and should not be interpreted as bandwidth-independent evidence. A small exception appears in SFX ViSQOL, which shows a slight improvement at higher bandwidth (e.g., 2.4 at 44.1/48 kHz versus 2.3 at 16/24 kHz). We interpret this cautiously: although overall separation fidelity declines with bandwidth, the additional high-frequency detail available at higher sampling rates may slightly improve the perceptual realism of certain broadband and transient-heavy SFX signals.

How do the bandwidth-scaled CodecSep variants compare with AudioSep? The bandwidth study is not only a scaling experiment; it also helps contextualize the sampling-rate mismatch with AudioSep. Against the **retrained 32 kHz AudioSep+dnr-v2** baseline ($-5.6/7.7/-4.5$ SI-SDR; 2.6/2.5/2.3 ViSQOL), **CodecSep at 24 kHz** still performs better on all three stems in SI-SDR and also improves ViSQOL for music and speech while matching SFX ViSQOL. Even the **44.1 kHz** and **48 kHz** CodecSep variants remain substantially stronger than retrained AudioSep on music and SFX SI-SDR, although their speech SI-SDR drops below the 16/24 kHz CodecSep variants.

Relative to the much stronger **pretrained 32 kHz AudioSep** checkpoint ($-2.5/4.9/-0.3$ SI-SDR; 2.9/3.1/2.6 ViSQOL), the comparison is more mixed. **CodecSep at 24 kHz** still exceeds pretrained AudioSep in SI-SDR on all three stems, while the **44.1 kHz** and **48 kHz** CodecSep variants continue to outperform it on speech SI-SDR and remain competitive on music/SFX SI-SDR. In ViSQOL, however, the pretrained AudioSep checkpoint remains especially strong perceptually, particularly for SFX, which is unsurprising given its much larger training corpus. These comparisons therefore suggest two things simultaneously: lower bandwidth does help CodecSep, but the underlying masking formulation remains viable beyond 16 kHz rather than collapsing outside the narrowband case.

Why do we not report a 32 kHz codec-domain counterpart? There is currently no publicly available neural audio codec in our setup operating at **32 kHz**, so an exactly matched codec-domain 32 kHz counterpart was not available for a cleaner apples-to-apples comparison. We therefore use the best available higher-

bandwidth codec backbones (24, 44.1, and 48 kHz) to test whether the core method still functions beyond the 16 kHz case. Importantly, all of these codec-based variants preserve the main deployment advantage of CodecSep: they can operate directly on code streams, including a codes-in / codes-out path, with minimal loss relative to the continuous-latent path.

What is the main takeaway? The main takeaway is twofold. First, the **16 kHz DAC model is used as the primary CodecSep system** because it best matches the practical low-MAC setting that motivates this work. Second, the bandwidth-scaling results show that the *same masking interface remains effective across multiple codec backbones and sampling rates*, even though performance degrades as bandwidth increases. The comparison to AudioSep should therefore be read as follows: AudioSep is benchmarked in its standard and strongest available **32 kHz** form, while CodecSep is benchmarked in the codec-native settings supported by available neural codec backbones, with **16 kHz** used as the primary low-MAC operating point and higher-bandwidth experiments included specifically to contextualize the sampling-rate mismatch. Our claim is therefore not that 16 kHz and 32 kHz are perfectly matched operating points, but that (i) CodecSep is effective in the codec-mediated deployment regime, (ii) its masking interface remains valid as bandwidth increases, and (iii) higher-bandwidth operation becomes progressively harder. We view these results as an initial step toward full-band, and eventually stereo/spatial, operation within the same masking framework.

4.1.9 Additional experiments (cf. Appendix F–H).

For readability, several extended studies are deferred to the Appendix, which also provides full details on data construction, prompt protocols (generic vs. universal and paraphrased variants), training/evaluation splits, and metrics.

Does CodecSep generalize beyond the training distribution (cf. Appendix F)? Yes. On AudioCaps (derived from AudioSet), the spectrogram baseline AudioSep benefits from stronger distributional alignment and attains the best absolute scores. Even so, CodecSep+dnr-v2 generalizes competitively, surpassing AudioSep+dnr-v2 in SI-SDR at comparable ViSQOL (cf. Table 11). When both systems are retrained on AudioCaps, CodecSep+AudioCaps again outperforms AudioSep+AudioCaps in separation quality. The same pattern carries over to the more challenging dnr-v2 test set, where mixtures often contain speech, music, and multiple overlapping SFX: both models degrade in absolute terms, but CodecSep retains its advantage in SI-SDR.

How consistent are the gains over AudioSep across datasets and prompt settings (cf. Appendix G)? To answer this, we report relative percent improvements of CodecSep over AudioSep under matched training data and prompt conditions. CodecSep shows large SI-SDR gains on dnr-v2, smaller but still positive gains under paraphrased prompts, and consistent improvements across ESC-50, Clotho-v2, AudioSet, VGGSound, and AudioCaps, with comparatively modest ViSQOL deltas (cf. Table 12). These summaries complement the absolute tables in the main paper by showing that the performance trend is not isolated to a single dataset or prompt configuration.

What do reconstruction diagnostics reveal about leakage and mixture consistency (cf. Appendix H)? We examine this using a *single-source reconstruction* diagnostic on dnr-v2, where each input contains one target source and models are either prompted with the matching caption (text-guided) or routed to the corresponding fixed head (non–text-guided). We also report *mixture reconstruction* by summing predicted source stems for a mixture and comparing the result to the original mixture. This is intended as a leakage/consistency analysis rather than as a primary separation benchmark; the full setup and results are provided in Appendix H (cf. Table 13).

4.1.10 Extension to multi-modal prompting.

Because conditioning enters only via a fixed-dimensional query embedding e_τ that drives FiLM in the masker, the architecture is agnostic to the prompt modality. Concretely, one can replace the text encoder with (i) an *audio* encoder to accept audio prompts $\left(e_\tau^{\mathrm{aud}}\right)$, (ii) an *image/vision–language* encoder (e.g., CLIP) to accept image prompts $\left(e_\tau^{\mathrm{vis}}\right)$, or (iii) a lightweight fusion (e.g., gated additive or attention pooling) of $\left(e_\tau^{\mathrm{text}}, e_\tau^{\mathrm{aud}}, e_\tau^{\mathrm{vis}}\right)$ to support mixed prompts—all without modifying the masker or the codec.

5 Conclusion

CodecSep advances text-guided universal sound separation by operating directly in NAC latents with a FiLM-conditioned Transformer masker (not a decoder), outperforming audio stream separators like AudioSep across dnr-v2 and five open-domain datasets (AudioCaps, ESC–50, Clotho–v2, AudioSet, VGGSound) under matched training/prompt protocols. In code-stream deployments, it needs just 1.35 GMACs end-to-end—$\sim$54$\times$ less compute (25$\times$ architecture-only) than spectrogram domain separators like AudioSep—while remaining fully bitstream-compatible. Ablations indicate that DAC latents are sufficiently structured that masking over them yields stronger separation than decoding/generating sources from the latents; a MOS–LQS study corroborates perceptual gains. We also demonstrate a codes in: codes out route that operates on reconstructed embeddings without fine-tuning, highlighting deployment readiness. We provide supplementary code to facilitate reproducibility.

6 Limitations and Clarifications.

We discuss the limitations of our work as follows–

(1) *Data and prompts.* Training data scale and prompt diversity are modest relative to open-domain audio. As shown in Table 5, finer SFX supervision sharpens SFX extraction *and* improves speech/music stems; larger, more heterogeneous corpora spanning multiple prompt granularities—including temporal/relational cues—should yield further gains.

(2) *Temporal prompting.* While CodecSep is robust to synonymic paraphrases, we did not evaluate prompts with explicit temporal structure (e.g., causal ordering), which remains an open direction.

(3) *Perceptual SFX quality.* In some settings, SFX perceptual quality trails the best competing scores despite superior SI-SDR; improving SFX naturalness without sacrificing separation is future work.

(4) *Channel-wise masking and residual overlap.* Our qualitative analysis suggests that the learned masks are predominantly channel-wise rather than strongly time-localized. While this is sufficient for effective separation in many cases, it also means that sources sharing partially overlapping latent channels may not be perfectly disentangled. As a result, some non-target sources can remain weakly active in the separated outputs, especially in dense mixtures with overlapping events. Addressing such residual overlap may require richer masking mechanisms that incorporate stronger temporal selectivity or more explicit source disentanglement objectives.

7 Future Work

Several directions could further strengthen codec-native universal sound separation.

Richer data and prompt supervision. A natural next step is to scale training to larger and more heterogeneous audio corpora with broader prompt diversity. In particular, training with mixtures of prompt granularities—from coarse category prompts to fine-grained compositional descriptions—may improve both generalization and controllability. Extending the prompt space to include temporal, relational, and referring-expression cues is also an important direction.

Beyond channel-wise masking. Our qualitative analysis suggests that CodecSep primarily relies on channel-wise latent reweighting. While effective, this also leaves residual overlap in dense mixtures where multiple sources share partially overlapping latent channels. Future work could therefore explore richer masking mechanisms that combine channel-wise modulation with stronger temporal selectivity, explicit cross-source competition, or additional disentanglement-oriented objectives.

Improved perceptual quality. Although CodecSep achieves strong SI-SDR and competitive ViSQOL, perceptual quality for some SFX conditions still trails the best competing systems. A promising direction is to incorporate auxiliary perceptual objectives, embedding-consistency losses, or lightweight refinement stages that improve naturalness without sacrificing the efficiency advantages of codec-domain separation.

Higher-bandwidth and spatial audio. Our bandwidth-scaling study suggests that the masking interface remains valid at higher sampling rates, but separation becomes more difficult as high-frequency and transient detail become more entangled. Future work should therefore investigate larger-capacity maskers, improved codec backbones, and training at 24 kHz, 44.1 kHz, and 48 kHz more systematically. Extending the framework to higher-bandwidth as well as stereo and spatial audio is another important direction, for example with 48 kHz stereo EnCodec, HO–DirAC Hold et al. (2024), or SpatialCodec Xu et al. (2024).

Multimodal prompting. Because conditioning enters only through a fixed-dimensional query embedding, CodecSep naturally admits extensions beyond text. Future work could study audio-guided, image-guided, or mixed multimodal prompting, allowing users to specify targets through reference sounds, images, or combined cues without changing the core masking architecture.

Deployment and on-device validation. While our current evidence supports the usefulness of CodecSep for codec-mediated edge/server pipelines through compute, memory, and latency analysis, direct deployment on mobile or embedded hardware remains to be demonstrated. An important future direction is therefore end-to-end validation on real on-device platforms, including power, latency, and memory measurements under realistic streaming workloads.

8 Broader Impact

CodecSep is motivated by a practical systems question: can universal, prompt-guided sound separation be made lightweight enough to operate in codec-mediated edge–server pipelines, rather than only in compute-heavy waveform or spectrogram domains? In that sense, the main positive impact of this work is not only improved separation quality under matched evaluation, but also a shift toward more deployment-realistic audio models. By operating directly on neural audio codec latents, CodecSep reduces compute, memory, and latency substantially in code-stream settings, which may broaden access to source-separation technology in resource-constrained environments. This could benefit assistive listening, hearing augmentation, speech enhancement in communication systems, low-bandwidth audio editing, interactive media production, and on-device or edge-based content manipulation where repeated decode–separate–re-encode cycles are undesirable.

A second potential positive impact is flexibility. Unlike fixed-stem separation systems that are restricted to a predefined output taxonomy, CodecSep supports prompt-guided extraction and therefore moves toward more open-vocabulary interaction with audio mixtures. In principle, this may make audio tools easier to control for non-expert users, since extraction can be specified semantically rather than through fixed source heads. The architectural result may also be of broader research interest: the paper suggests that modern neural audio codec latents already contain enough source-structured information for masking-based extraction, which may encourage more efficient codec-native approaches for related audio understanding and generation tasks.

At the same time, such flexibility introduces risks. Prompt-guided extraction can be misused for privacy-invasive or surveillance-oriented applications, such as isolating speech, speakers, or background events from recorded mixtures without the knowledge or consent of those being recorded. More generally, any method that lowers the computational barrier for source separation can also lower the barrier for extracting sensitive content from audio. We do not claim that CodecSep solves speaker identification, diarization, or forensic enhancement, but the ability to isolate semantically specified content could still be misapplied in ways that raise privacy concerns.

There are also risks of misuse in media manipulation. Improved sound separation can facilitate unauthorized remixing, decontextualization, or repurposing of copyrighted or personal audio content. In creative settings this may be beneficial for editing and accessibility, but in adversarial settings it could enable misleading edits, selective removal of contextual sounds, or extraction of material that creators did not intend to be

isolated. Because CodecSep is text-conditioned, an additional concern is that users may over-trust the semantic controllability of the system and assume that a prompt uniquely identifies a source, even when prompts are ambiguous or multiple events overlap.

Several properties of the present work limit these risks but do not eliminate them. First, the model is evaluated primarily in mono, prompt-guided source extraction rather than speaker-level forensic recovery. Second, the paper explicitly shows that performance degrades under paraphrased or ambiguous prompts and at higher bandwidths, and that perceptual quality for some SFX settings remains imperfect. Third, the strongest efficiency gains arise in codec-mediated deployments, so the system is most naturally suited to applications where codec infrastructure already exists. These limitations mean that CodecSep should not be interpreted as a turnkey solution for arbitrary audio surveillance or perfect open-world extraction. Nevertheless, even imperfect systems can be misused, especially if integrated into larger pipelines.

We therefore view responsible deployment as important. In practical applications, safeguards could include clear user-facing disclosure that separated outputs are model-generated estimates rather than ground-truth stems, access controls for sensitive domains, provenance logging for edited audio, and restrictions on use in privacy-sensitive or consent-critical settings. Benchmarking should also expand beyond fidelity to include misuse-relevant evaluations such as prompt ambiguity, failure under overlapping sources, and robustness against extracting unintended content.

Finally, the environmental impact of this work is mixed. Training modern audio models still consumes non-trivial compute, and our experiments use GPU resources. However, one motivation of CodecSep is to reduce inference-time cost in realistic deployment settings. If such codec-native systems replace heavier decode–separate–re-encode pipelines at scale, they may reduce operational energy usage during inference, especially in repeated or latency-sensitive edge–server workflows. We therefore believe the broader impact of this work is potentially positive, provided that efficiency gains are paired with clear communication of limitations and with responsible use in privacy- and consent-sensitive contexts.

9 Declaration of LLM Usage.

LLM is used only to aid or polish writing and does not impact the core methodology, scientific rigorousness, or originality of the research.

References

Xiaoyu Bie, Xubo Liu, and Gaël Richard. Learning source disentanglement in neural audio codec. *arXiv preprint arXiv:2409.11228*, 2024.

Zalán Borsos, Raphaël Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matt Sharifi, Dominik Roblek, Olivier Teboul, David Grangier, Marco Tagliasacchi, and Neil Zeghidour. Audiolm: a language modeling approach to audio generation, 2023. URL https://arxiv.org/abs/2209.03143.

Honglie Chen, Weidi Xie, Andrea Vedaldi, and Andrew Zisserman. Vggsound: A large-scale audio-visual dataset. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 721–725. IEEE, 2020a.

Jingjing Chen, Qirong Mao, and Dong Liu. Dual-path transformer network: Direct context-aware modeling for end-to-end monaural speech separation. *arXiv preprint arXiv:2007.13975*, 2020b.

Michael Chinen, Felicia S. C. Lim, Jan Skoglund, Nikita Gureev, Feargus O'Gorman, and Andrew Hines. Visqol v3: An open source production ready objective speech and audio metric, 2020. URL https://arxiv.org/abs/2004.09584.

Michaël Defferrard, Kirell Benzi, Pierre Vandergheynst, and Xavier Bresson. Fma: A dataset for music analysis, 2017. URL https://arxiv.org/abs/1612.01840.

Alexandre Défossez, Nicolas Usunier, Léon Bottou, and Francis R. Bach. Music source separation in the waveform domain. *CoRR*, abs/1911.13254, 2019. URL http://arxiv.org/abs/1911.13254.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. URL https://arxiv.org/abs/1810.04805.

Konstantinos Drossos, Samuel Lipping, and Tuomas Virtanen. Clotho: An audio captioning dataset. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 736–740. IEEE, 2020.

Zhihao Du, Jiaming Wang, Qian Chen, Yunfei Chu, Zhifu Gao, Zerui Li, Kai Hu, Xiaohuan Zhou, Jin Xu, Ziyang Ma, Wen Wang, Siqi Zheng, Chang Zhou, Zhijie Yan, and Shiliang Zhang. Lauragpt: Listen, attend, understand, and regenerate audio with gpt, 2024. URL https://arxiv.org/abs/2310.04673.

Alexandre Défossez, Nicolas Usunier, Léon Bottou, and Francis Bach. Music source separation in the waveform domain, 2021. URL https://arxiv.org/abs/1911.13254.

Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. High fidelity neural audio compression, 2022. URL https://arxiv.org/abs/2210.13438.

Eduardo Fonseca, Xavier Favory, Jordi Pons, Frederic Font, and Xavier Serra. Fsd50k: An open dataset of human-labeled sound events, 2022. URL https://arxiv.org/abs/2010.00475.

Jort F. Gemmeke, Daniel P. W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *Proc. IEEE ICASSP 2017*, New Orleans, LA, 2017.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In Yee Whye Teh and Mike Titterington (eds.), *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pp. 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR. URL https://proceedings.mlr.press/v9/glorot10a.html.

Romain Hennequin, Anis Khlif, Felix Voituret, and Manuel Moussallam. Spleeter: a fast and efficient music source separation tool with pre-trained models. *Journal of Open Source Software*, 5(50):2154, 2020.

Christoph Hold, Leo McCormack, Archontis Politis, and Ville Pulkki. Perceptually-motivated spatial audio codec for higher-order ambisonics compression, 2024. URL https://arxiv.org/abs/2401.13401.

Yanxin Hu, Yun Liu, Shubo Lv, Mengtao Xing, Shimin Zhang, Yihui Fu, Jian Wu, Bihong Zhang, and Lei Xie. Dccrn: Deep complex convolution recurrent network for phase-aware speech enhancement. *arXiv preprint arXiv:2008.00264*, 2020.

Ilya Kavalerov, Scott Wisdom, Hakan Erdogan, Brian Patton, Kevin Wilson, Jonathan Le Roux, and John R. Hershey. Universal sound separation. In *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 175–179, 2019. doi: 10.1109/WASPAA.2019.8937253.

Chris Dongjoo Kim, Byeongchang Kim, Hyunmin Lee, and Gunhee Kim. Audiocaps: Generating captions for audios in the wild. In *NAACL-HLT*, 2019.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. URL https://arxiv.org/abs/1412.6980.

Rithesh Kumar, Prem Seetharaman, Alejandro Luebs, Ishaan Kumar, and Kundan Kumar. High-fidelity audio compression with improved rvqgan. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 27980–27993. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/58d0e78cf042af5876e12661087bea12-Paper-Conference.pdf.

Jonathan Le Roux, Scott Wisdom, Hakan Erdogan, and John R Hershey. Sdr–half-baked or well done? In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 626–630. IEEE, 2019.

Kai Li, Runxuan Yang, and Xiaolin Hu. An efficient encoder-decoder architecture with top-down attention for speech separation, 2023. URL https://arxiv.org/abs/2209.15200.

Xubo Liu, Qiuqiang Kong, Yan Zhao, Haohe Liu, Yi Yuan, Yuzhuo Liu, Rui Xia, Yuxuan Wang, Mark D. Plumbley, and Wenwu Wang. Separate anything you describe, 2024. URL https://arxiv.org/abs/2308.05037.

Yi Luo and Nima Mesgarani. Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(8):1256–1266, August 2019. ISSN 2329-9304. doi: 10.1109/taslp.2019.2915167. URL http://dx.doi.org/10.1109/TASLP.2019.2915167.

Tanvir Mahmud, Saeed Amizadeh, Kazuhito Koishida, and Diana Marculescu. Weakly-supervised audio separation via bi-modal semantic similarity. *arXiv preprint arXiv:2404.01740*, 2024.

Koyel Mukherjee, Alind Khare, and Ashish Verma. A simple dynamic learning rate tuning algorithm for automated training of dnns, 2019. URL https://arxiv.org/abs/1910.11605.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: An asr corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5206–5210, 2015. doi: 10.1109/ICASSP.2015.7178964.

William Peebles and Saining Xie. Scalable diffusion models with transformers, 2023. URL https://arxiv.org/abs/2212.09748.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018. doi: 10.1609/aaai.v32i1.11671. URL https://ojs.aaai.org/index.php/AAAI/article/view/11671.

Darius Petermann, Gordon Wichern, Zhong-Qiu Wang, and Jonathan Le Roux. The cocktail fork problem: Three-stem audio separation for real-world soundtracks. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 526–530, 2022. doi: 10.1109/ICASSP43922.2022.9746005.

Karol J. Piczak. ESC: Dataset for Environmental Sound Classification. In *Proceedings of the 23rd Annual ACM Conference on Multimedia*, pp. 1015–1018. ACM Press. ISBN 978-1-4503-3459-4. doi: 10.1145/2733373.2806390. URL http://dl.acm.org/citation.cfm?doid=2733373.2806390.

Jordi Pons, Xiaoyu Liu, Santiago Pascual, and Joan Serrà. Gass: Generalizing audio source separation with large-scale data. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 546–550, 2024. doi: 10.1109/ICASSP48485.2024.10446601.

Daniel Stoller, Sebastian Ewert, and Simon Dixon. Wave-u-net: A multi-scale neural network for end-to-end audio source separation. *arXiv preprint arXiv:1806.03185*, 2018.

Cem Subakan, Mirco Ravanelli, Samuele Cornell, Mirko Bronzi, and Jianyuan Zhong. Attention is all you need in speech separation. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 21–25. IEEE, 2021.

Naoya Takahashi, Nabarun Goswami, and Yuki Mitsufuji. Mmdenselstm: An efficient combination of convolutional and recurrent neural networks for audio source separation. In *2018 16th International workshop on acoustic signal enhancement (IWAENC)*, pp. 106–110. IEEE, 2018.

Efthymios Tzinis, Zhepei Wang, Xilin Jiang, and Paris Smaragdis. Compute and memory efficient universal sound source separation. *Journal of Signal Processing Systems*, 94(2):245–259, 2022a.

Efthymios Tzinis, Gordon Wichern, Aswin Subramanian, Paris Smaragdis, and Jonathan Le Roux. Heterogeneous target speech separation. *arXiv preprint arXiv:2204.03594*, 2022b.

Efthymios Tzinis, Gordon Wichern, Paris Smaragdis, and Jonathan Le Roux. Optimal condition training for target source separation. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023.

Emmanuel Vincent, Tuomas Virtanen, and Sharon Gannot. *Audio source separation and speech enhancement*. John Wiley & Sons, 2018.

Chengyi Wang, Sanyuan Chen, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, Lei He, Sheng Zhao, and Furu Wei. Neural codec language models are zero-shot text to speech synthesizers, 2023. URL `https://arxiv.org/abs/2301.02111`.

Xiaofei Wang, Manthan Thakker, Zhuo Chen, Naoyuki Kanda, Sefik Emre Eskimez, Sanyuan Chen, Min Tang, Shujie Liu, Jinyu Li, and Takuya Yoshioka. Speechx: Neural codec language model as a versatile speech transformer, 2024. URL `https://arxiv.org/abs/2308.06873`.

Scott Wisdom, Efthymios Tzinis, Hakan Erdogan, Ron Weiss, Kevin Wilson, and John Hershey. Unsupervised sound separation using mixture invariant training. *Advances in neural information processing systems*, 33: 3846–3857, 2020.

Yusong Wu, Ke Chen, Tianyu Zhang, Yuchen Hui, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, 2023. doi: 10.1109/ICASSP49357.2023.10095969.

Zhongweiyang Xu, Yong Xu, Vinay Kothapally, Heming Wang, Muqiao Yang, and Dong Yu. Spatialcodec: Neural spatial speech coding, 2024. URL `https://arxiv.org/abs/2309.07432`.

Jia Qi Yip, Chin Yuen Kwok, Bin Ma, and Eng Siong Chng. Speech separation using neural audio codecs with embedding loss, 2024a. URL `https://arxiv.org/abs/2411.17998`.

Jia Qi Yip, Shengkui Zhao, Dianwen Ng, Eng Siong Chng, and Bin Ma. Towards audio codec-based speech separation, 2024b. URL `https://arxiv.org/abs/2406.12434`.

Dong Yu, Morten Kolbæk, Zheng-Hua Tan, and Jesper Jensen. Permutation invariant training of deep models for speaker-independent multi-talker speech separation. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 241–245. IEEE, 2017.

Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. Soundstream: An end-to-end neural audio codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30: 495–507, 2022. doi: 10.1109/TASLP.2021.3129994.

A   Failure Modes of PIT and MixIT for Universal Sound Separation

Permutation-Invariant Training (PIT) and Mixture-Invariant Training (MixIT) have historically been effective for closed-domain separation tasks where the number of underlying sources is known, fixed, or varies within a narrow, well-defined range. However, their underlying assumptions lead to structural limitations when applied to open-domain universal source separation (USS), where mixtures may contain an arbitrary and potentially large number of heterogeneous sound events. In this section, we summarize the key failure modes observed when training with PIT/MixIT to extend fixed-stem models to open-domain mixtures.

A core limitation of PIT/MixIT is the requirement to specify a maximum number of output sources, denoted by N. During training, the model produces exactly N outputs for every mixture, and the PIT or MixIT objective establishes a correspondence between these outputs and the underlying reference sources (or intermediate MixIT partitions). This design is brittle in scenarios where the true number of sources varies widely. When the mixture contains more than N sources—a frequent occurrence in open-domain audio—the model has no mechanism to create additional outputs. Instead, it suffers source collision and collapses multiple sources into a single output stem, resulting in unavoidable leakage, loss of fine structure, and a sharp degradation in separation quality. The post-hoc identification step cannot recover the missing sources, because the model never produced separate representations for them in the first place; those sources simply do not exist within the model's output space.

Conversely, when the mixture contains fewer than N sources, the model is still obligated to return N outputs. This mismatch introduces new problems: several outputs correspond to no actual source and become "inactive" stems, while others may capture residual background energy or hallucinated content. These false positives degrade metrics such as SI-SDR and create ambiguity during evaluation because the model does not encode which stems are meant to be meaningful. Such outputs also make deployment difficult, as downstream systems must decide which stems to trust and which to ignore.

The reliance on a fixed maximum number of sources N also places a heavy burden on both training stability and computational cost. As N increases, the permutation space in PIT expands combinatorially, and MixIT assignments become increasingly complex, making training slow, unstable, and in many cases prone to divergence. In open-domain datasets such as dnr-v2, mixtures may contain eight or more concurrent sound events, forcing PIT/MixIT baselines to adopt impractically large values of N to avoid source collisions. In practice, such configurations are computationally prohibitive and empirically unreliable.

These limitations collectively illustrate why PIT and MixIT, despite their historical success in speech separation and other closed-set tasks, are poorly suited for open-domain universal separation. Their fixed-output architecture is fundamentally mismatched to real-world mixtures that contain highly variable and unpredictable numbers of sources. In contrast, CodecSep bypasses this bottleneck entirely through free-form text-guided inference: the model extracts only the requested source category, emits no unused stems, and scales naturally to mixtures with arbitrary levels of overlap. This flexibility enables CodecSep to support both closed-set and open-domain use cases, while also providing a foundation for future extensions to fine-grained extraction of individual speaker stems, instrument stems, or sound-effect stems.

B Design Rationale: FiLM-Conditioned Masking in NAC Latent Space

Problem setup and pipeline contrast. Let $x(t) \in \mathbb{R}$ be a mono mixture with sources $\{y_s(t)|s \in \mathcal{S}\}$, where $x(t)$ is expressed as $x(t) = \sum_{s \in \mathcal{S}} y_s(t)$.

Spectrogram-domain (AudioSep):

$$x \xrightarrow{\text{STFT}} X \in \mathbb{C}^{F \times T_{spec}} \xrightarrow{g(X, e_\tau)} \tilde{Y}_s = |\hat{M}_s| \odot |X| \exp(\angle X + \angle \hat{M}_s) \xrightarrow{\text{ISTFT}} \tilde{y}_s(t),$$

where $g(\cdot, e_\tau)$ denotes a FiLM-conditioned, complex-domain spectrogram separator that predicts a magnitude mask $|\hat{M}_s| \in [0,1]^{F \times T_{spec}}$ and a phase residual $\angle \hat{M}_s$, conditioned jointly on the mixture spectrogram X (obtained by the STFT of x) and the text embedding e_τ. The predicted magnitude mask and phase residual are then applied element-wise to form the source spectrogram $\tilde{Y}_s$ which is subsequently transformed back to the time domain via ISTFT to obtain $\tilde{y}_s(t)$.

CodecSep (NAC latent-domain):

$$x \xrightarrow{Enc(\cdot)} Z \in \mathbb{R}^{d \times T} \xrightarrow{Mask(Z, e_\tau)} \tilde{Z}_s = M_s \odot Z \xrightarrow{Dec(\cdot)} \tilde{y}_s(t),$$

with a frozen DAC backbone $Enc(\cdot), Dec(\cdot)$, and a FiLM-conditioned transformer masker $Mask(\cdot, e_\tau)$ that estimates a soft mask $M_s \in [0,1]^{d \times T}$ conditioned on both the codec latents $Z = Enc(x)$ and the text embedding e_τ. The predicted mask M_s is applied element-wise to Z to produce source-specific latents $\tilde{Z}_s = M_s \odot Z$, which are subsequently decoded by $Dec(\cdot)$ to obtain the separated waveform $\tilde{y}_s(t)$.

Dimensionality reduction and compression. Operating on NAC latents Z slashes dimensionality while preserving perceptual factors. For 1 s audio at 32 kHz, complex STFT with $N=1024$ and hop size $M=320$ samples has $T_{spec} \approx 100$ frames and $F = 2 \times 1024$ (Re+Im) scalars per frame, so $F \cdot T_{spec} \approx 204{,}800$. A 16 kHz DAC with width $d=64$ and the same M yields $T \approx 50$ and $d \cdot T = 64 \times 50 = 3{,}200$ ($\sim 64\times$ smaller), shrinking $Q/K/V$ and MLP sizes and easing self-attention. Similarly, for 32 kHz NACs like EnCodec, $T \approx 50$ with $d=128$, so attention/MLPs still operate on $\sim 32\times$ smaller latents than complex STFTs.

B.1 Why NAC latents (vs. spectrograms) and how the codec prior enables separation?

STFT vs. NAC encoding. The STFT is a *linear* projection from $x(t)$ to $X \in \mathbb{C}^{F \times T}$ and does not explicitly preserve or organize the intrinsic semantic structure of audio. Consequently, spectrogram-based separators (e.g., AudioSep) require an additional learned encoder–decoder (typically CNN/ResUNet) to first compress X into high-dimensional latent features and then decode spectrogram masks that are discriminatively trained for separation. This couples *semantic abstraction* and *separation* into one network, increasing parameters and MACs and forcing the model to learn structure "from scratch" in a redundant, noisy representation.

NAC encoder as a semantic prior. Neural audio codecs (DAC) are trained with perceptual, adversarial, and codebook objectives that encourage the encoder $Enc(\cdot)$ to map x into compact, structured latents:

$$Enc(.) : x \mapsto Z \in \mathbb{R}^{d \times T}.$$

These latents lie on a *discriminative compressed manifold* $\mathcal{M}_{\text{latent}}$ in which semantically meaningful factors (e.g., pitch, timbre, transients) are disentangled and aligned for downstream use. In our system, we operate directly on the *continuous* latents Z (from Enc), use a FiLM-conditioned transformer masker to predict a separation mask M_k from a textual query τ_k, and form the estimate

$$\tilde{Z}_k = M_k \odot Z, \qquad \tilde{y}_k = Dec(\tilde{Z}_k),$$

thereby leveraging the codec's structured manifold for masking instead of learning a new representation.

Separation mapping in latent vs. spectrogram space. In CodecSep, the separator learns

$$Mask(.,.) :\ Z \in \mathcal{M}_{\text{latent}} \rightarrow \hat{Z}_k,$$

which is easier to optimize because the input is denoised, compressed, and semantically organized by the codec. In contrast, spectrogram models must learn

$$g(.,.): \; X \in \mathbb{C}^{F \times T} \; \to \; \hat{Y}_k,$$

over a noisier, higher-dimensional space *without* semantic compression.

Hierarchical (RVQ) structure that benefits separation. The codec applies Residual Vector Quantization (RVQ) to Z, producing discrete codes $A = [a_t \in [K]^{N_q}]_{t=1}^T$ with $K = 1024$ and N_q quantizers. Codebook lookup yields embeddings

$$e_t \; = \; \sum_{i=1}^{N_q} \mathrm{lookup}(a_t^{(i)}), \qquad E = [e_t | t \in [T]] \approx Z,$$

and $Dec(E)$ reconstructs the waveform. The RVQ cascade induces a natural coarse-to-fine hierarchy: the first quantizer captures coarse structure (e.g., low-frequency content, speaker/instrument timbre, global acoustic traits), while later quantizers refine residual details (e.g., high-frequency components, onsets/transients, background textures). This hierarchy mirrors the discriminative cues needed for separation and is directly exploitable by a mask-based transformer.

Loss-induced organization of the latent space. The DAC objective shapes Z using complementary terms:

- *Multi-scale spectral loss* $\mathcal{L}_{\mathrm{mel}}$ to preserve perceptually relevant frequency content at multiple time scales;

- *Time-domain reconstruction loss* $\mathcal{L}_{\mathrm{feat}} = \|x(t) - \tilde{y}(t)\|_1$ for fidelity and stability;

- *Multi-resolution adversarial loss* $\mathcal{L}_{\mathrm{adv}}$ with (i) multi-period waveform discriminators (pitch/periodicity) and (ii) multi-band STFT discriminators (fine spectral detail), plus a *feature-matching* term $\mathcal{L}_{\mathrm{feat}}^G$;

- *Codebook loss* $\mathcal{L}_{\mathrm{code}}$ to ensure compact, diverse, well-utilized codes and reinforce RVQ's coarse-to-fine disentanglement.

- *Quantizer dropout (RVQ stage dropout)*: randomly disabling a subset of RVQ stages during training to discourage over-reliance on late codebooks and encourage smoother coarse-to-fine residual allocation, yielding more bitrate-robust and well-structured RVQ representations.

The overall loss is

$$\mathcal{L}_{\mathrm{DAC}} \; = \; \lambda_{\mathrm{mel}} \mathcal{L}_{\mathrm{mel}} \; + \; \lambda_{\mathrm{feat}} \mathcal{L}_{\mathrm{feat}} \; + \; \lambda_{\mathrm{adv}} \mathcal{L}_{\mathrm{adv}} \; + \; \lambda_{\mathrm{code}} \mathcal{L}_{\mathrm{code}},$$

yielding latents that are (i) *denoised* (robust to low-level artifacts), (ii) *semantic* (preserve pitch, timbre, temporal structure), (iii) *disentangled* (coarse-to-fine RVQ), and (iv) *efficient* (bitrate-aware constraints).

Separation benefits from the codec prior (and contrast to spectrogram baselines). Because Z is already semantically organized, the FiLM-conditioned masker operates on a representation that encodes the right factors for selection, not generation. This (i) reduces compute and memory (the separator acts on compact Z instead of X), (ii) accelerates convergence and improves stability (masking over a clean manifold), and (iii) improves robustness to prompt variation (text FiLM modulates semantically aligned channels). In contrast, spectrogram systems must learn a task-specific latent from X and perform separation *jointly*, which increases parameter count and MACs, slows convergence, and can lead to overfitting in the absence of the codec's inductive bias.

B.2 Transformer-based masker vs. decoder-style generation (specific design choice).

In CodecSep, we replace decoder-style source generation (as in CodecFormer, which directly predicts the target waveform/spectrogram) with a *Transformer-based, FiLM-conditioned masker* that outputs a soft mask over the codec latents. Concretely,

$$M_s \; = \; Mask(Z, e_\tau) \in [0,1]^{d \times T}, \qquad \tilde{Z}_s \; = \; M_s \odot Z, \qquad \tilde{y}_s(t) \; = \; Dec(\tilde{Z}_s).$$

This choice has the following concrete advantages (all in the NAC latent domain):

- **Efficient and stable training.** Predicting M_s to *modulate* existing latent content is a simpler, more constrained learning problem than end-to-end *generation* of $\tilde{Z}_s$ or $\tilde{y}_s(t)$ with a decoder head. Working in Z avoids the instability commonly observed in direct waveform/spectrogram prediction, leading to faster convergence and lower training variance under the same optimization settings.

- **Direct leverage of codec-disentangled latents.** DAC/RVQ pretraining organizes Z into a coarse-to-fine, semantically disentangled manifold (periodicity, timbre, transients). The masker exploits this structure by *gating along* these axes to isolate sources, rather than using Z merely as input to a decoder that *generates* new latents/waveforms. This turns separation into selection on a well-structured space, improving identifiability while reducing parameters and data demands.

- **Minimal distortion through modulation (no hallucination).** The separator does not synthesize new content; it rescales and selects what is already encoded in Z. Forming $\tilde{Z}_s = M_s \odot Z$ preserves the mixture's latent structure and reduces artifacts relative to encoder–decoder separation pipelines that generate source signals from scratch, thereby limiting hallucinations and leakage.

- **Preservation of long-term temporal/spectral structure.** The NAC encoder has already organized periodicity, timbre, and transient structure in Z. Masking retains this organization across long contexts, whereas fully convolutional decoders trained to *generate* sources often exhibit long-term inconsistencies (e.g., drift over time or loss of periodic cues) when reconstructing from scratch.

- **Efficient use of Transformer capacity.** Instead of *synthesizing* source signals, the masker learns to *gate* semantically organized channels in Z, which is a substantially lighter optimization problem than decoder-style generation. FiLM conditioning steers the transformer to decide *where/how much* information to pass—not *what* to generate—so parameters and compute in the $Q/K/V$ and MLP projections are concentrated on selection and attenuation.

B.3 Why We Integrated FiLM Conditioning into NAC-Based Separation

We deliberately integrate *Feature-wise Linear Modulation (FiLM)* Perez et al. (2018) *inside* the transformer masker to inject text semantics while preserving the codec manifold and keeping $Enc(\cdot)/Dec(\cdot)$ frozen.

Targeted placement (masker, mid-layers). Given a CLAP text embedding e_τ, a lightweight query network $query(\cdot)$ produces per-layer affine parameters

$$(\gamma^l, \beta^l)_{l=1}^L \;=\; query(e_\tau), \quad \gamma^l, \beta^l \in \mathbb{R}^d,$$

which modulate intermediate activations $H^l \in \mathbb{R}^{d \times T}$ for $l \in \{2, \ldots, L-1\}$:

$$\tilde{H}_s^l \;=\; \text{FiLM}\big(H^l; \gamma^l, \beta^l\big) \;=\; \gamma^l \odot H^l + \beta^l.$$

Placing FiLM in the *masker* (not in *Enc* or *Dec*) keeps the codec latent distribution intact and confines conditioning to the *selection* step.

Lightweight computation (overhead and parameterization). FiLM adds only small per-layer vectors (γ^l, β^l) and a compact $query(\cdot)$ MLP; it does *not* increase sequence length, attention heads, or the quadratic attention cost. The extra FLOPs/params are negligible relative to multi-head attention and MLP blocks, aligning with CodecSep's efficiency goals.

Non-iterative inference (single forward pass). FiLM applies in a single pass through the masker. Unlike iterative conditioning mechanisms (e.g., flow-matching-based sampling), there are *no* sampling steps, thus preserving low latency for edge and hybrid deployments.

Manifold preservation and stability. Because FiLM scales/shifts existing channels of H^l rather than rewriting Z or generating $\tilde{Z}_s$ from scratch, the NAC manifold structure (periodicity, timbre, transients) is preserved. Empirically this reduces training variance and mitigates long-horizon inconsistencies common with generator-style heads.

B.4 Why report results on Z (continuous latents) and how to extend to code streams $A \to E$

Why we evaluate on continuous latents Z. We choose to perform—and therefore report—separation on the *continuous* DAC encoder latents $Z = Enc(x) \in \mathbb{R}^{d \times T}$ for the following concrete reasons:

1. **End-to-end gradient flow for separation.** Our training updates only the masker and query networks while keeping the codec frozen. Using Z allows straightforward backpropagation through $Mask(\cdot, e_\tau)$ and $Dec(\cdot)$ without dealing with discrete indices or straight-through estimators; gradients are well-behaved and convergence is consistently stable.

2. **Representational fidelity and disentanglement.** During codec pretraining, the RVQ cascade *regularizes Z* so that pitch, timbre, onsets/transients, and background textures are cleanly and hierarchically organized. Z therefore provides a richer, more disentangled signal for FiLM-conditioned masking than hard code indices, which are subject to quantization coarsening.

3. **Training stability and variance.** In our setting (frozen codec), operating on Z avoids variability from codebook utilization dynamics (e.g., late-stage RVQ sensitivity, bitrate truncation). Empirically, this reduces run-to-run variance and removes the need for specialized regularizers when training the separator.

How to extend the same model to discrete code streams. For deployment scenarios where the input is a compressed bitstream, we operate on the *reconstructed embeddings E* obtained from the codes A via codebook lookup, and train the masker on E instead of Z:

$$A = [a_t \in [1024]^{N_q} \mid t \in [T]], \qquad e_t = \sum_{i=1}^{N_q} \text{lookup}\big(a_t^{(i)}\big), \tag{14}$$

$$E = [e_t]_{t=1}^{T} \approx Z, \qquad \tilde{E}_s = M_s \odot E, \qquad \tilde{y}_s(t) = Dec(\tilde{E}_s) \tag{15}$$

Compressed bitstream path (codes-in, codes-out). Given $E \approx Z$ and the element-wise masking operation, the estimated source embeddings satisfy

$$\tilde{E}_s = M_s \odot E \approx M_s \odot Z = Z_s$$

When a uniform codec pathway (or codes-out interface) is required, we re-quantize the masked embeddings and optionally decode via the codec:

$$\hat{A}_s = Quant(\tilde{E}_s), \qquad \hat{E}_s = \text{lookup}(\hat{A}_s), \qquad \tilde{y}_s(t) = Dec\big(\hat{E}_s\big)$$

In deployments that only need to return a bitstream, the server can emit $\hat{A}_s$ directly and defer decoding to the client; otherwise, decoding $Dec(\tilde{E}_s)$ or $Dec(\hat{E}_s)$ yields the waveform on-device or server-side, respectively.

Embedding alignment (CodecFormer-EL Yip et al. (2024a)). To tighten $\tilde{E}_s \approx Z_s$, we can optimize an embedding-level consistency loss as in CodecFormer-EL Yip et al. (2024a)instead of our SI-SDR objective:

$$\mathcal{L}_{\text{emb}} = \sum_s \|\tilde{E}_s - Z_s\|_1$$

Why this works. By design of the codec, E *approximates Z* at the operating bitrate; the decoder already reconstructs $\hat{x}(t) = Dec(E)$ with high fidelity. Since our separator is a *masker* (selective modulation) rather than a generator, replacing Z with E preserves the semantics needed for separation while enabling

direct operation on bitstreams. In practice, the extension amounts to training (or fine-tuning) the same FiLM-conditioned transformer on E with unchanged objectives.

Summary of scope. We present results on Z to (i) isolate separator performance without discrete-index training complications, (ii) leverage continuous gradient flow and stable optimization, and (iii) align FiLM conditioning with a smooth latent manifold. The deployment-ready path to discrete code streams is immediate: switch the masker input to E via lookup from A and train/fine-tune accordingly, with optional distribution alignment and bitrate-robustness augmentation.

B.5 Deployment advantages of CodecSep over spectrogram-domain separators (AudioSep)

Central motivation. In realistic deployments, edge devices already run a neural audio codec; they transmit *code streams* rather than raw waveforms. CodecSep treats the codec backbone as part of the separation stack and operates *directly* on codec representations, thereby eliminating redundant *decode → separate → re-encode* cycles required by spectrogram-domain systems.

Pipeline comparison (server-side separation).   *Traditional (spectrogram) pipeline:*

$$\text{Edge: } \underbrace{Enc(.)}_{\text{codec}} \Rightarrow \text{Code} \Rightarrow \text{Server: } \underbrace{Dec(.) + g(\cdot, e_\tau) + Enc(.)}_{\text{decode + separate + re-encode}} \Rightarrow \text{Code} \Rightarrow \text{Edge: } Dec(.)$$

CodecSep pipeline:

$$\text{Edge: } \underbrace{Enc(.)}_{\text{codec}} \Rightarrow \text{Code} \Rightarrow \text{Server: } \underbrace{Mask(\cdot, e_\tau)}_{\text{mask on } E \approx Z} \Rightarrow \text{Code} \Rightarrow \text{Edge: } Dec(.)$$

CodecSep performs FiLM-modulated masking in the codec latent domain and returns separated code streams for edge-side decoding; spectrogram systems must decode to waveform (or magnitude/phase), separate in X, and re-encode.

Complexity accounting.   Let C_{enc} and C_{dec} be codec encode/decode costs, C_{spec} the spectrogram separator cost, and C_{mask} the CodecSep masker cost.

> **Code-stream input (typical):** AudioSep: $C_{\text{dec}} + C_{\text{spec}} + C_{\text{enc}}$,   CodecSep: C_{mask}.
> **Audio-stream input (edge-only):** AudioSep: C_{spec},   CodecSep: $C_{\text{enc}} + C_{\text{mask}} + C_{\text{dec}}$.

In the common (code-stream) case, CodecSep removes both decode and re-encode on the server. Moreover, *within* the separator, CodecSep operates on Z/E with $|Z| \ll |X|$ (cf. Sec. B), reducing activation memory and bandwidth throughout attention and MLP blocks.

Interface compatibility with codec bitstreams.   When only quantized codes $A = [a_t | t \in [T]]$ are available, we reconstruct embeddings by codebook lookup $E = [e_t | t \in [T]]$ with $e_t = \sum_{i=1}^{N_q} \text{lookup}(a_t^{(i)}) \approx Z$ and apply the same masker:

$$A \Rightarrow E \approx Z \xrightarrow{Mask(\cdot, e_\tau)} \tilde{E}_s \Rightarrow \text{Code stream out.}$$

No architectural change is required; separation remains in the codec latent domain and stays fully compatible with streaming/edge ecosystems.

Why spectrogram-domain systems incur extra overhead.   Spectrogram separators (e.g., AudioSep) are defined on $X = \text{STFT}(x)$. Given code-stream inputs, they must first run $Dec(.)$ to obtain a waveform, compute X, perform separation, and then run $Enc(.)$ to return codes. This decode + separate + re-encode loop adds latency, memory traffic, and energy cost on the server path and scales poorly with concurrent streams.

Operational advantages of CodecSep.

- **Eliminates redundant codec cycles in server workflows:** With code streams, we reconstruct embeddings $E \approx Z$ by codebook lookup (Sec. B.4), avoiding server-side decode/re-encode; only the edge decodes the final stems.

- **Smaller working representation during separation:** masking in Z/E (e.g., $d{=}64$) reduces intermediate activations, lowering memory bandwidth and enabling tighter batching.

- **Non-iterative, single-pass conditioning:** FiLM-conditioning within the masker adds negligible overhead and preserves low latency (no iterative sampling).

- **Seamless edge/server hybrid and edge-only modes:** identical separator logic serves both modalities; with code-stream inputs, the server path remains *separator-only.*

- **Maintains codec manifold structure:** by modulating Z/E rather than rebuilding X, CodecSep preserves periodicity/timbre/transients already organized by the codec, supporting stable, high-fidelity stems at deployment.

C   Dataset Details

C.1   Divide and Remaster v 2.0 (dnr-v2)

dnr-v2 Petermann et al. (2022) dataset consists of $60s$-duration artificial mixtures of speech, music, and SFX sampled from LibriSpeech Panayotov et al. (2015), Free Music Archive (FMA) Defferrard et al. (2017), and Freesound Dataset $50K$ (FSD50K) Fonseca et al. (2022), respectively. It includes $3,406$ ($56.7hrs$) training, 487 ($8.13hrs$) validation, and 973 ($16.22hrs$) test mixtures, each provided with its three individual source audios. The mixtures are generated by normalizing each source to fixed Loudness Units Full-Scale (LUFS) levels: -17 dB (speech), -24 dB (music), and -21 dB (SFX), with ±2 dB random perturbations. Any source exceeding a peak threshold is normalized to 0.5 dB. The sources are mixed and normalized to -27 dB LUFS with additional random perturbations. The validation and test sets are trimmed for silence and split into $5s$ or $10s$ segments. Segments where sources are present for less than 50% of the duration are removed, resulting in $2,852$ ($\approx 3.96hrs$) validation and $1,840$ ($\approx 5.11hrs$) test mixtures.

While originally developed for 3-stem separation, we adapt dnr-v2 to the USS setting by replacing fixed source labels with natural language descriptions. For speech or music stem, we use broad, category-level prompts (e.g., "speech," "music"), reflecting realistic usage in production workflows. In contrast, SFX sources are more complex—often containing three or more overlapping events. We generate prompts to query the SFX stem using FSD50K's hierarchical annotations, combining fine-grained class labels with their parent categories. This results in long-form, compositional queries that reflect the structure of the mixture (e.g., "dog barking, Animal, engine rumbling, motor vehicle").

C.2   Open-Domain Benchmarks

We benchmark on five open-domain datasets spanning captioned audio, environmental sounds, and large multi-event corpora: AudioCaps Kim et al. (2019), an AudioSet-derived collection of > 46k $10s$ YouTube clips paired with human-written captions describing the dominant sound events (used by us to synthesize training and test mixtures); ESC-50 Piczak, a curated environmental sound dataset of 2,000 clips ($5s$ each) organized into 50 classes with 40 examples per class across five meta-categories (animals, natural, human non-speech, domestic, exterior/urban); Clotho-v2 Drossos et al. (2020), 6,974 audio samples (15–$30s$) each annotated with five human captions (8–20 words) covering open-domain events; AudioSet Gemmeke et al. (2017), the evaluation split of AudioSet comprising human-labeled $10s$ YouTube clips over an ontology of 632 audio event classes in a multi-label setting; and VGGSound Chen et al. (2020a), an AudioSet-derived audio–visual corpus with $550+$ hours of $10s$ segments covering a wide variety of everyday sound categories. For AudioCaps we form both training and testing mixtures (same scale of test data as dnr-v2) by summing three clips (validation segmented into $5s$, test preserves clips up to $20s$), while for ESC-50, Clotho-v2, AudioSet-eval, and VGGSound we construct test-only mixtures using the same three-clip protocol.

Figure 4: Evaluation workflow for dnr-v2. Each mixture contains multi-source stems: speech (often multi-speaker), music (multi-instrument), and SFX (≥ 3 overlapping events). Fixed-stem baselines predict a fixed set of outputs (e.g., 3 stems), whereas CodecSep and other text-guided models generate only the prompted source. Speech and music are evaluated using generic prompts, while SFX uses long-form compositional prompts listing all SFX events in each mixture. Extracted signals are compared with ground-truth category stems using SI-SDR and ViSQOL.

D   Evaluation Details

D.1   Divide and Remaster v 2.0 (dnr-v2)

The dnr-v2 benchmark presents a challenging open-domain separation setting: although the dataset provides three category labels—speech, music, and sound effects—each category represents a multi-source stem. A single mixture frequently contains five to ten underlying acoustic sources, including overlapping speakers, multiple musical instruments, and several sound-effect events occurring either concurrently or in sequence. The three reported stems are therefore semantic groupings that support interpretability and reproducibility, rather than an indication that the mixture contains only three sources. Any evaluation methodology must respect this structure. Figure 4 provides an overview of our dnr-v2 evaluation workflow and highlights how these multi-source stems are handled across fixed-stem un-guided and text-guided models.

For fixed-stem unguided architectures, evaluation is performed by mapping each predicted output stem to one of the three ground-truth stems and computing SI-SDR and ViSQOL on a per-category basis. Importantly, we do not employ PIT or MixIT training objectives for these baselines; instead, we train dedicated three-stem models that directly predict speech, music, and SFX stems.

Text-guided models, including CodecSep, follow a fundamentally different inference and evaluation paradigm. Speech and music stems are recovered using generic prompts ("speech," "music"), which reliably capture their multi-source content. In contrast, SFX stems require mixture-specific prompts because sound effects span a wide and open-domain label space. For each mixture, we use a long-form compositional prompt enumerating all SFX events present in the ground truth. This ensures that the model has sufficient semantic context to extract the full SFX stem. The number of sfx events present in a mixture varies considerably. We additionally perform an ambiguous-prompt evaluation, where deliberately underspecified prompts for speech and music are used to assess robustness to vague or incomplete semantic queries. After inference, the extracted waveform for each category is directly compared with the corresponding ground-truth stem using SI-SDR and ViSQOL. This evaluation design ensures fairness between fixed-stem and text-guided systems while faithfully reflecting the multi-source structure of dnr-v2 mixtures

Figure 5: Evaluation workflow for the standardized three-source benchmarks (AudioCaps, ESC-50, Clotho, VGGSound, and AudioSet-eval). Following prior USS protocols, each mixture is constructed by combining three isolated events drawn from distinct classes. For each class, the corresponding textual prompt is supplied to the separator (e.g., "dog barking," "gun shot," "motor vehicle"), and the extracted signal is compared with the ground-truth isolated source using SI-SDR and ViSQOL.

D.2 Open-Domain Benchmarks

For AudioCaps, ESC-50, Clotho, VGGSound, and AudioSet-eval, we adopt the standardized three-source mixture protocol. Following established practice, each mixture is constructed by combining three isolated events drawn from different classes. Each source is then extracted by the text-guided models using its corresponding textual caption as prompt, and evaluation metrics are computed against the ground-truth isolated audio. Figure 5 illustrates the evaluation workflow used for these benchmarks, highlighting how class-specific prompts are applied and how the resulting predictions are matched against the ground-truth isolated sources. Although these datasets do not reflect the complex multi-source structure of dnr-v2, the standardized 3-way protocol enables direct benchmarking against prior work (AudioSep) under consistent conditions.

E   Training Details

The complete model, including the query module $query(.)$, is trained for $400K$ iterations with DAC Kumar et al. (2023) and CLAP Wu et al. (2023) modules frozen. Validation is conducted every $5K$ iterations and test every $10K$ iterations. We use ADAM Kingma & Ba (2017) as our optimizer and train with a batch size of 4 examples, each 2 seconds in duration, and a learning rate of $1.5e^{-4}$ on a single 24GB NVIDIA A-30 GPU. Training employs a $ReduceLRonPlateau$ Mukherjee et al. (2019) scheduler, which reduces the learning rate by a factor of 0.5 if the validation loss does not improve for two consecutive validation checks. We train two versions of CodecSep, one using the dnr-v2 dataset and the other using AudioCaps, to evaluate performance across different training distributions. We refer to these models using the suffixes +dnr-v2 and +AudioCaps, respectively, to indicate which dataset each model was trained on.

Since TDANet and CodecFormer were originally designed for speech separation, we re-train newly initialized 3-stem versions on the dnr-v2 training set using the same configuration as CodecSep. We also train a 3-stem Sudo rm-rf model to compare against compute-efficient separators. For AudioSep, we evaluate both the publicly available pretrained model—trained on diverse datasets—and versions re-trained on dnr-v2 and AudioCaps for consistency. We include SDCodec using the official pretrained checkpoints released by the authors. Finally, we incorporate the USS-pretrained variant of BiModalSS and re-train SudoRmRf+FiLM on dnr-v2 with CLAP text conditioning. To ensure a fair comparison, all inputs to TDANet, Sudo rm-rf, AudioSep, BiModalSS, and Sudo rm-rf+FiLM undergo codec processing with a full-band stereo-capable $48\,\mathrm{kHz}$ EnCodec during training and inference. This accounts for codec-induced distortions and artifacts, reflecting realistic deployment scenarios where audio is typically processed through compression pipelines in cloud-based systems.

Table 11: Generalization and transfer results for universal sound separation.

(a) Generalization on **AudioCaps-test**.

Model	Separation	
	SI-SDR ($\uparrow$)	**ViSQOL** ($\uparrow$)
AudioSep	$\mathbf{-2.5^{\pm 12.14}}$	$\mathbf{2.4^{\pm 1.08}}$
AudioSep + dnr-v2 (zero-shot)	$-6.4^{\pm 11.48}$	$2.3^{\pm 1.08}$
CodecSep + dnr-v2 (zero-shot)	$\mathbf{-6.1^{\pm 11.62}}$	$2.2^{\pm 1.16}$
AudioSep + AudioCaps	$-9.2^{\pm 18.71}$	$2.3^{\pm 1.11}$
CodecSep + AudioCaps	$\mathbf{-6.2^{\pm 10.58}}$	$2.1^{\pm 1.00}$

(b) Transfer to **dnr-v2-test** when trained on AudioCaps (zero-shot on dnr-v2).

Model	Metric ($\uparrow$)	Music	Speech	Sfx
AudioSep + AudioCaps	SI-SDR	$-14.9^{\pm 23.08}$	$-7.1^{\pm 25.80}$	$-14.6^{\pm 23.26}$
(zero-shot)	ViSQOL	$2.4^{\pm 0.71}$	$2.4^{\pm 0.70}$	$2.2^{\pm 0.79}$
CodecSep + AudioCaps	SI-SDR	$-8.5^{\pm 2.78}$	$2.5^{\pm 2.91}$	$-5.9^{\pm 4.33}$
(zero-shot)	ViSQOL	$2.3^{\pm 0.53}$	$2.6^{\pm 0.47}$	$2.1^{\pm 0.72}$

F   Cross-Benchmark Performance: AudioCaps and dnr-v2.

Table 11a reports generalization results on the AudioCaps-test set derived from AudioSet. AudioSep benefits from distributional alignment, having been trained on diverse datasets, including AudioSet, and consequently achieves the strongest separation performance. CodecSep+dnr-v2 generalizes well to AudioCaps and outperforms AudioSep+dnr-v2 in SI-SDR while maintaining competitive ViSQOL scores. When retrained on AudioCaps, CodecSep again outperforms AudioSep in separation quality, demonstrating strong cross-domain robustness. Table 11b further supports this trend on the more challenging dnr-v2 test set, where CodecSep+AudioCaps outperforms AudioSep+AudioCaps in SI-SDR across all sources while maintaining comparable perceptual quality. However, both models experience a performance drop on dnr-v2 due to its increased mixture complexity, often containing speech, music, and three or more overlapping SFX sources—making it significantly more challenging than the simpler mixtures seen during AudioCaps training.

Table 12: Relative gains (%) of **CodecSep** over **AudioSep** under matched training/prompt settings. Each sub-table reports percent improvements for a specific evaluation setup.

(a) DnR-v2 test set

Metric	Relative Gain (%)		
	Speech	Music	SFX
SI-SDR	+29.8	+120.7	+119.1
ViSQOL	+26.1	+10.5	+0.5

(b) Ambiguous prompts (speech & music paraphrases)

Metric	Relative Gain (%)	
	Speech	Music
SI-SDR	+1.2	+13.0
ViSQOL	+3.8	+1.2

(c) Additional open-domain benchmarks

Metric	AudioCaps	ESC-50	Clotho-v2	AudioSet	VGGSound
SI-SDR	+5.5	+24.3	+30.0	+16.4	+13.0
ViSQOL	−4.3	+2.2	+2.4	+2.9	+2.9

(d) Training on AudioCaps

Metric	AudioCaps-test	dnr-v2		
		Music	Speech	SFX
SI-SDR	+32.5	+43.1	+134.7	+59.5
ViSQOL	−6.5	−3.8	+5.4	-4.2

G   Discussion of Relative–Gain Summaries.

Tables 12a–12d consolidate relative improvements of CodecSep over AudioSep under matched training data and prompt protocols, complementing the absolute results in the main text. On dnr-v2 (cf. Table 12a), CodecSep delivers large SI-SDR gains—especially for music and SFX—together with a clear perceptual lift. Under paraphrased prompts (cf. Table 12b), gains are smaller but remain positive, indicating robustness to lexical variation. Across additional open-domain benchmarks (cf. Table 12c), SI-SDR gains are consistent while ViSQOL deltas are modest, aligning with cross-domain trends reported earlier. Finally, when trained on AudioCaps (cf. Table 12d), CodecSep maintains an advantage on AudioCaps-test and yields strong improvements on dnr-v2, supporting the claim that codec-latent masking generalizes well across datasets and prompt regimes.

Table 13: Results: Reconstruction Performance, Universal Sound Separation (**dnr-v2-test**)

Model	Metric (↑)	Reconstruction			
		Mixture	**Music**	**Speech**	**Sfx**
3-Stem: Fixed Stem, Non Text-guided					
TDANet	SI-SDR	$-3.3^{\pm 7.78}$	$\mathbf{8.0^{\pm 5.29}}$	$\mathbf{11.1^{\pm 3.32}}$	$\mathbf{4.7^{\pm 5.11}}$
	ViSQOL	$3.9^{\pm 0.35}$	$\mathbf{4.2^{\pm 0.46}}$	$\mathbf{4.5^{\pm 0.32}}$	$\mathbf{4.1^{\pm 0.39}}$
CodecFormer	SI-SDR	$-47.6^{\pm 9.51}$	$-47.1^{\pm 10.97}$	$-47.8^{\pm 9.65}$	$-48.2^{\pm 9.87}$
	ViSQOL	$1.0^{\pm 0.07}$	$1.0^{\pm 0.12}$	$1.0^{\pm 0.06}$	$1.2^{\pm 0.47}$
CodecSep + dnr-v2	SI-SDR	$3.4^{\pm 1.85}$	$4.1^{\pm 3.97}$	$6.2^{\pm 2.87}$	$0.8^{\pm 5.16}$
(unguided, 3-stem)	ViSQOL	$3.2^{\pm 0.20}$	$3.0^{\pm 0.33}$	$3.5^{\pm 0.24}$	$3.2^{\pm 0.46}$
SDCodec	SI-SDR	$\mathbf{7.0^{\pm 2.49}}$	$7.7^{\pm 4.60}$	$8.3^{\pm 3.26}$	$2.5^{\pm 5.65}$
	ViSQOL	$\mathbf{4.3^{\pm 0.15}}$	$4.0^{\pm 0.28}$	$4.4^{\pm 0.15}$	$4.0^{\pm 0.34}$
Text-guided					
AudioSep	SI-SDR	$5.5^{\pm 1.96}$	$4.7^{\pm 5.36}$	$11.0^{\pm 2.99}$	$-2.0^{\pm 5.68}$
(zero-shot)	ViSQOL	$4.1^{\pm 0.38}$	$3.8^{\pm 0.65}$	$4.6^{\pm 0.13}$	$3.2^{\pm 0.77}$
AudioSep + dnr-v2	SI-SDR	$6.5^{\pm 2.26}$	$8.0^{\pm 4.55}$	$8.1^{\pm 3.35}$	$2.3^{\pm 5.95}$
	ViSQOL	$4.2^{\pm 0.18}$	$4.1^{\pm 0.21}$	$3.0^{\pm 0.29}$	$3.8^{\pm 0.47}$
CodecSep + dnr-v2	SI-SDR	$4.1^{\pm 2.06}$	$3.9^{\pm 3.93}$	$6.1^{\pm 2.86}$	$0.7^{\pm 5.29}$
	ViSQOL	$3.7^{\pm 0.22}$	$3.4^{\pm 0.33}$	$3.8^{\pm 0.24}$	$3.5^{\pm 0.44}$
CodecSep + dnr-v2	SI-SDR	$\mathbf{12.2^{\pm 2.42}}$	$\mathbf{12.6^{\pm 3.81}}$	$\mathbf{13.6^{\pm 2.59}}$	$\mathbf{8.7^{\pm 4.17}}$
(ablate **Masker**)	ViSQOL	$\mathbf{4.4^{\pm 0.14}}$	$\mathbf{4.1^{\pm 0.31}}$	$\mathbf{3.9^{\pm 0.34}}$	$3.8^{\pm 0.54}$
AudioSep + AudioCaps	SI-SDR	$6.7^{\pm 2.52}$	$8.1^{\pm 4.63}$	$8.4^{\pm 3.21}$	$2.4^{\pm 6.12}$
(zero-shot)	ViSQOL	$4.2^{\pm 0.19}$	$4.1^{\pm 0.21}$	$4.2^{\pm 0.21}$	$\mathbf{3.8^{\pm 0.46}}$
CodecSep + AudioCaps	SI-SDR	$0.6^{\pm 1.89}$	$-0.2^{\pm 5.15}$	$-11.^{\pm 5.21}$	$1.2^{\pm 4.84}$
(zero-shot)	ViSQOL	$3.3^{\pm 0.23}$	$2.9^{\pm 0.64}$	$1.7^{\pm 0.48}$	$3.4^{\pm 0.41}$

H   Further Studies: Reconstruction Performance.

Table 13 assesses performance under a single-source reconstruction setting on the dnr-v2-test set, where each model is prompted to reproduce the input source. In addition, we report mixture reconstruction scores obtained by summing the separated sources and comparing them to the original mixture.

Among non-text-guided models, TDANet yields the best single-source reconstruction, while SDCodec performs better on mixture reconstruction. Replacing decoder-style generation in CodecFormer with a Transformer masker over codec latents in CodecSep+ dnr-v2, (unguided 3-stem) markedly improves both per-stem and mixture reconstruction fidelity and perceptual quality. Masking modulates information already organized in the codec manifold (Z) rather than re-synthesizing it from scratch, avoiding collapse/artifacts and yielding tighter mixture consistency.

Among the text-guided models, CodecSep achieves reconstruction performance comparable to the pre-trained and retrained AudioSep variants across all source types. AudioSep consistently excels in reconstruction due to its STFT-based masking pipeline, which enables more controlled and artifact-free waveform synthesis. However, CodecSep surpasses the pretrained AudioSep in SFX reconstruction—across both SI-SDR and ViSQOL on dnr-v2. Notably, the masker ablated CodecSep variant delivers the best reconstruction performance on dnr-v2. With isolated single-source input and matching prompts, direct conditioning minimally disturbs the NAC latent space, allowing high-fidelity reconstruction in this variant. However, strong mixture reconstruction despite poor separation suggests source leakage across separated outputs.