DEFORMER: LATENT-BASED MUSIC SOURCE SEPARATION BEYOND FOUR STEMS

Anonymous authors
Paper under double-blind review

ABSTRACT

In music source separation, we aim to decompose a stereo music recording into a set of individual audio tracks, each isolating a specific instrument or group of instruments. Currently, most systems separate a given mixture considering a four-stem convention: drums, bass, vocals and a residual other that absorbs every other instrument. In this work, we introduce DeFormer, a music source separation model capable of separating a given mixture to 12 stems, decomposing at the latent representation level. Expanding the task from 4 to 12 demands greater sensitivity to instrument-specific cues due to finer distinctions in the target distribution, e.g., sparsity, spectral (or timbral) entanglement, etc. DeFormer addresses these challenges by combining a novel latent-separation architecture with a training recipe adapted to sparse and skewed supervision. On our held-out set, DeFormer reaches 6.59 dB mean cSDR over 12 targets, 1.06 dB ahead of the closest of four baselines we retrain at the same target count. DeFormer also demonstrates a notable gap in subjective performance on both general audio quality and adjacent source bleeding, measured in a conducted human study involving domain-expert annotators with professional music production backgrounds. Code and model checkpoints will be released upon acceptance. Samples page: https://deformerpaper-dev.github.io/DeFormer

1 INTRODUCTION

Modern music is commonly produced in layers, where a recording is built from separate tracks of grouped instruments (stems), balanced and summed into the single stereo mixture. Music source separation aims to retrieve the decomposition by recovering the stems from the mixture (Huang et al., 2014). Such separation enables applications in music information retrieval (Gómez et al., 2018), transcription (Manilow et al., 2020), music education (Cano et al., 2014) and generative modeling (Tal et al., 2024; Parker et al., 2024). Nowadays, music source separation systems are mostly built and evaluated against the traditional four-stem taxonomy: bass, drums, vocals and everything else grouped under "other" (Rouard et al., 2023; Lu et al., 2024; Tong et al., 2024; Fabbro et al., 2024).

This partition reflects the structure of the data rather than the intended taxonomy. Clean, aligned, per-instrument multi-tracks are difficult to source and rarely public. MoisesDB (Pereira et al., 2023), one of the largest datasets in the field supporting a target taxonomy beyond the standard 4-stem setting, contains 240 tracks. While valuable for model evaluation, its limited scale makes it insufficient for training state-of-the-art models. The 4-stem partition is also a simpler one from acoustic perspective, as vocals, drums and bass occupy distinct spectral and temporal characteristics, while a real mix's guitars, keys and woodwinds collapse into *other*, a placeholder rather than a source.

To explore task expansion at a larger data scale, we annotated $\sim$ 15k stemmed songs, which revealed three difficulties that become more acute beyond four stems (Sec. 3). (i) *The targets are entangled.* Unlike the four-stem classes, finer families share spectral and timbral properties, so the dominant error shifts from missed energy to bleeding between slots; e.g. *sound-effects* leaking into *drums*. (ii) *The labels are uncertain, and not exclusive.* Naming an isolated stem's family is subjective, not measurement: three independent expert annotators returned identical labels on only 56.4% of stems, and some tracks carry two families at once, e.g. sitar and percussion. (iii) *Supervision is sparse and skewed.* Three stems appear in $\geq 70\%$ of songs, while all others appear only in 40% or under, with the rarest stem in $< 3\%$; so some slots could collapse to silence prediction.

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

An alternative line replaces fixed-output separation with query-conditioned extraction, using one network to return one requested target at a time (Lee et al., 2019; Chen et al., 2022). Watcharasupat & Lerch (2024) extend this approach with a single stem-agnostic bandsplit decoder conditioned on an audio-query embedding. The taxonomy then lives in the query and the architecture is tailored to conditional retrieval, so nothing ties the returned stems to one another.

In this work we introduce DeFormer, a music source separation model that separates a given music mixture into 12 stems. Entanglement drives the design. DeFormer performs separation in the bottleneck latent rather than in signal space. A linear map expands the encoded mixture into 12 streams, which attend to one another along time and frequency. Each stream then returns to audio through a shared decoder with a per-slot affine output layer. In our proposed training scheme, every mixture is constructed around a uniformly sampled anchor, multi-label stems are scored against the sum of their corresponding output slots, and the objective penalizes leakage into absent slots. For the primary 12-stem comparison, we retrain all baselines on the same corpus and training budget. DeFormer outperforms all four baselines retrained for 12 targets, improving mean cSDR over the best-performing baseline by 1.06 dB on our held-out set and by 0.58 dB on MoisesDB (Sec. 5.2).

A blind listening study with 80 domain experts with professional music-production experience further shows that DeFormer achieves the highest mean ratings among the evaluated systems for both audio quality and source isolation. In a sequential ablation with the objective held fixed, our proposed Fusion and De-Mix components improve mean cSDR by 1.30 dB over the $348M$-parameter scaled SCNet baseline while using fewer than half as many parameters (Sec. 5.4). Finally, we present a targeted error analysis of sparse stem classes provides insight into the data challenges introduced by the expanded taxonomy.

2 Related work

Four stems became the standard target set for reasons of data and convention rather than of method. The convention was not there at the start: music source separation began as unsupervised spectrogram factorization, splitting a magnitude spectrogram into non-negative spectral components and activations with no notion of a named instrument (Virtanen, 2007). Supervised deep models replaced the factorization with a learned time-frequency mask, first for singing-voice extraction (Huang et al., 2014). Convolutional and recurrent variants followed, whose gains came as much from augmentation and blending as from architecture (Uhlich et al., 2017). Two things then fixed the shape of the field. MUSDB18 supplied a common corpus of full tracks with four stems: vocals, drums, bass, and everything else lumped into *other* (Rafii et al., 2017; 2019). The separation evaluation campaigns then turned SDR into the reporting convention (Stöter et al., 2018; Fabbro et al., 2024).

Architecturally the work split into two lineages that merged only recently. Waveform-domain models discarded the spectrogram, from a one-dimensional U-Net over raw samples (Stoller et al., 2018) through masking in a learned basis (Luo & Mesgarani, 2019) to Demucs (Défossez et al., 2019). Spectrogram-domain models kept the STFT and pushed on the network instead, from recurrent baselines (Stöter et al., 2019) and widely used pre-trained four-stem models (Hennequin et al., 2020) to dense blocks and U-Nets that factorise time-frequency structure explicitly (Takahashi & Mitsufuji, 2020; Choi et al., 2020). The two lineages then converged: KUIELab-MDX-Net blends a time-frequency branch with a time-domain one (Kim et al., 2021), Hybrid Demucs runs both branches in parallel and sums them inside the network (Défossez, 2021), and HTDemucs replaces the innermost layers of each with a cross-domain transformer (Rouard et al., 2023).

More recently the spectrogram side has led again. It splits the spectrum into subbands and models them with recurrent (Luo & Yu, 2023) or rotary-attention (Lu et al., 2024; Wang et al., 2023) networks, or compresses the bands at unequal rates through a convolutional encoder–decoder (Tong et al., 2024). This family reports the strongest four-stem results published on MUSDB18-HQ, and these are the systems we compare against.

All of them allocate one output path per target in signal space. The work that goes beyond four targets keeps that allocation, requesting one stem at a time through a query or a label (Chen et al., 2022; Watcharasupat & Lerch, 2024; Watcharasupat et al., 2024). That choice makes the stem count a property of inference rather than of the model, and leaves nothing tying the requested stems to

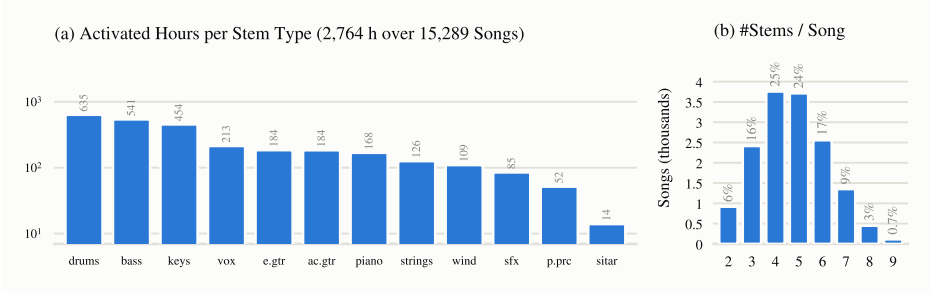

Figure 2: Skewed distribution of audio hours differing by order of magnitude; most songs have < 6 stems. **(a)** Total *activated* hours of the 12 stem types. **(b)** Stems per song; none contributes ≥ 10.

each other. We depart from it: all targets are emitted in one pass from a decoder shared across slots, separated within a shared latent representation, not by repeated conditioned inference.

3 DATA

Large-scale datasets for stem separation remain scarce. MoisesDB (Pereira et al., 2023), comprising 240 songs, enables 12-way evaluation but is insufficient for large-scale training. We therefore annotate $\sim$ 15k songs to expand the task from 4 to 12 stems (App.A). In the following we examine the challenges arising from this expansion, which motivate our design choices.

The targets are spectrally entangled. The standard bass, drums, vocals, and *other* partition is not merely driven by data availability; these targets exhibit less spectral and timbral overlap. Fig. 1 shows cosine similarities between pre-prediction-head embeddings of a single-stem classifier trained on our data. The standard 4 stems are the most distinct, whereas every additional stem has at least one counterpart with similarity ≥ 0.45. This entanglement, also reflected in the difficulty experienced by professional music producers in distinguishing stems, motivates our design choices.

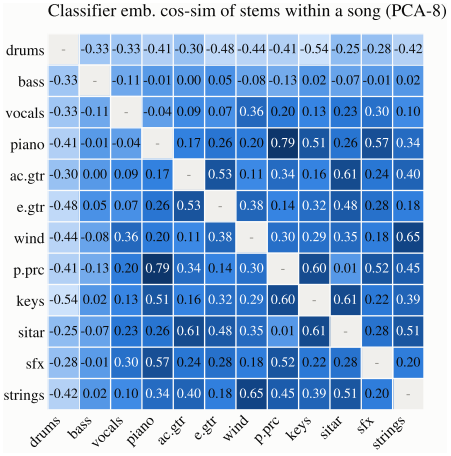

Figure 1: **Most targets are similar.** Mean pairwise cos-sim measured in 8 dim. PCA projected classifier embedding. Across the tested PCA dimensionalities, drums and bass remain the most isolated.

Labels are uncertain and multi-valued. Three professional music producers independently annotated each stem (App. B). Full 3/3 consensus occurred in only 56.4% of cases, further highlighting the ambiguity between stems. Relaxing consensus to 2/3 retains 94.0% of examples instead of discarding 43.6% of the data. Moreover, stems may contain multiple sources (e.g., sitar and percussion). We therefore retain labels agreed upon by at least two annotators, $\mathcal{Y} = \cup_{i \neq j} A_i \cap A_j$, where A_j denotes annotator j's labels, naturally supporting multi-label targets.

Supervision is sparse and skewed. As expected, most songs contain bass and drums while the presence of pitch percussion or woodwinds is notably lower; Fig 2a draws the distribution of activated (non-silent) audio hours in our corpus. In addition, as seen in Fig 2b, 49% of the songs in our corpus only have 4 or 5 stems in their mix, with less than 4% having 8 or more, so mixtures that include multiple stems are themselves rare, and the stems that do appear together fall into a few recurring combinations (App. C).

Batch sampling during training. In an attempt to address sparsity of the data, we build each training example around a sampled anchor. The anchor is sampled uniformly across stems to ensure that rare stems are present in expectation in every training batch, selecting a segment to be sampled from the *activated* regions of the anchor. To create the *mix* input we then sample all other stems of

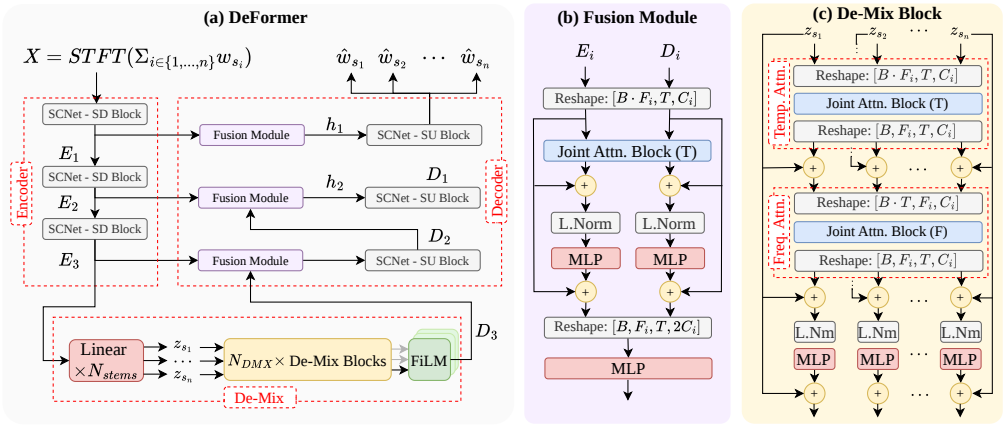

Figure 3: **DeFormer. (a)** Overview of the suggested model. Separation happens at the latent-level in the De-Mix module. The decoder is shared across slots but for a single affine output layer, and each stem latent sequence is decoded independently. **(b)** Fusion module architecture. **(c)** A De-Mix block attends over $m = K$ slot streams, once along time and once along frequency.

that song that are *active* in the sampled region, so the number of stems present in the *mix* follows the recording. In addition, with probability $p_{\text{mix}} = 10\%$ we additionally fill slots the song leaves empty by randomly sampling *active* segments from our corpus.

4 Method

4.1 Problem Setup

Let $\omega \in \mathbb{R}^{f_s d \times c}$ denote an audio signal of duration d seconds, sampled at rate f_s with c channels. A mixture is the sum of K stems, and the task is to invert that sum: given only ω, the model f_θ aims to recover estimates as close as possible to the sources that produced it,

$$\omega = \sum_{i\in\{1,\dots,K\}} \omega_{s_i}, \qquad (1) \qquad\qquad f_\theta : \omega \longmapsto (\hat{\omega}_{s_1}, \dots, \hat{\omega}_{s_K}), \qquad (2)$$

with ω, ω_{s_i}, $\hat{\omega}_{s_i} \in \mathbb{R}^{f_s d \times c}$, each stem a waveform holding one or more instrument recordings that jointly compose the song.

The model separates at the *latent* level: it emits one z_{s_i} per slot and returns each to the waveform domain as $\hat{\omega}_{s_i} = \text{Dec}_i(z_{s_i})$, where $\text{Dec}_i = \text{SU}_i \circ S$ composes a trunk S shared by every slot with a single affine output layer SU_i of its own. We add an *approximate additivity* secondary objective: decoding the sum of stem latents should return the mix, $\|\frac{1}{K}\sum_i \text{Dec}_i(\sum_{j\in\{1,\dots,K\}} z_{s_j}) - \omega\|_1 \to 0$.

4.2 Architecture

DeFormer (Fig. 3a) encodes an input mixture spectrogram using sparse down-sampling (SD) blocks, expands it into K latent streams in a *De-Mix* module, and walks back up, joining each encoder skip to its decoder stream in a *Fusion* module and passing it's output via sparse up-sampling (SU) blocks and an inverse STFT to return the waveforms at the top, see blocks description in App. D.2.

In our model design we focus on improving spectral modeling to mitigate the finer and inherently more entangled target taxonomy of our setup. We aim to perform the separation itself over the bottleneck latent, taking advantage of the extracted features obtained by the encoder SD blocks. To do so, we use a shared decoder module which operates identically for each stem, with the exception of a single per-stem affine layer at the end, allowing each slot its own gain at a low amount of added parameters (0.17% of model parameters). Including these layers allows extended freedom to focus on specific frequency ranges for a finer reconstruction per stem, following SCNet's design.

Building from SCNet, we take SD and SU blocks unchanged and replace the fusion and separation modules with our *Fusion* and *De-Mix* modules.

Both our suggested modules are built around joint attention blocks. Each joint attention module (Fig. 4a) attends m sequences (h_i) at once while keeping them distinct, and output m sequences (o_i). Attention is the only shared operation, so they are told apart by their own weights and position. We extend RoPE (Su et al., 2024) to carry two positions (Fig. 4b): (i) flat along the token axis, and (ii) per sequence positions, resetting at each boundary, resulting in an aligned relative position encoding in a cross-source manner.

In De-Mix (Fig. 3c) a linear map expands the deepest encoder output into K streams, so every slot keeps its own projections and feed-forward weights, and each block attends along time and along frequency. In Fusion (Fig. 3b), the decoder stream and the adjacent encoder skip attend over time alone with frequency folded into the batch. As skips are shared by every slot, attention is where a stream draws the details relevant to it.

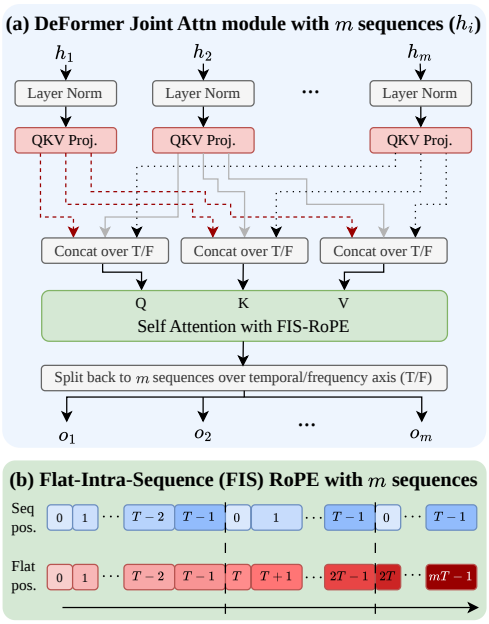

Figure 4: **(a)** Joint attention over m seq., attending to one another while keeping their own Norm and projections. **(b)** FIS-RoPE positions: per-seq. and flat along a token axis.

4.3 Training Objective

With architecture focused on improving spectral modeling, and cSDR being the main objective evaluation metric - we focus our main optimization goal on reconstruction, emphasizing spectral losses, and add regularization terms to discourage bleeding and silence prediction in rare stem bins.

4.3.1 Reconstruction Terms

We denote the present stem slots that combine the mix as $\mathcal{P} \subseteq \{1, \dots, K\}$, and the paired set over present stems as $\mathcal{X}(\mathcal{P}) = \{(\hat{\omega}_i, \omega_i)\}_{i \in \mathcal{P}}$. Similarly, we denote $\mathcal{A}$ for the complement of $\mathcal{P}$. We let α_L^i be a scale hyperparameter that varies per loss L and stem i. The first of two reconstruction terms is the mean absolute error in the waveform domain; and the second supervises the spectral envelope, comparing power mel spectrograms at the single resolution $n_0 = 1024$ over $F = 64$ bands, with $P_n(u)$ the mel power spectrogram of u:

$$\ell_1\big(\mathcal{X}(\mathcal{P})\big) = \sum_{i,(\hat{u},u)\in\mathcal{X}(\mathcal{P})} \frac{\alpha_{\ell_1}^i}{|\mathcal{P}|}\|\hat{u}-u\|_1 \quad \ell_{\text{mel}}\big(\mathcal{X}(\mathcal{P})\big) = \sum_{i,(\hat{u},u)\in\mathcal{X}(\mathcal{P})} \frac{\alpha_{\text{mel}}^i}{|\mathcal{P}|}\big\|P_{n_0}(\hat{u})-P_{n_0}(u)\big\|_1, \quad (3)$$

where $\|\cdot\|_1$ is the mean absolute value over all axes and $(\hat{u}, u)$ is any estimate–reference pair. To further emphasize the spectral component we adopt the multi-resolution spectral loss (Défossez et al., 2023) and swap the mel spectrograms with complex STFTs; taken over window lengths $\mathcal{N} = \{512, 1024, 2048\}$, writing $\Delta_n = X_n(\hat{u}) - X_n(u)$ for the difference of the complex STFTs and $\Delta^{\text{re}}, \Delta^{\text{im}}$ for its real and imaginary parts,

$$\ell_{\text{cstft}}\big(\mathcal{X}(\mathcal{P})\big) = \frac{1}{|\mathcal{P}|} \sum_{i,(\hat{u},u)\in\mathcal{X}(\mathcal{P})} \left[\frac{\alpha_{\text{cstft}}^i}{|\mathcal{N}|} \sum_{n\in\mathcal{N}} \sqrt{\frac{1}{CF_nT_n} \sum_{c,f,t} \left((\Delta_{n,cft}^{\text{re}})^2 + (\Delta_{n,cft}^{\text{im}})^2 \right)} \right]. \quad (4)$$

Then we could summarize the full reconstruction term as,

$$\mathcal{L}_{\text{recon}} = \ell_1\big(\mathcal{X}(\mathcal{P})\big) + \ell_{\text{mel}}\big(\mathcal{X}(\mathcal{P}')\big) + \ell_{\text{cstft}}\big(\mathcal{X}(\mathcal{P}')\big); \mathcal{P}' \subseteq \mathcal{P}. \quad (5)$$

Where $\mathcal{P}'$ is a subset of the present stems set. To further dis-encourage zero prediction for the less frequent stems, we increase their relative weight α_L^i in the loss components. The spectral terms dictate most of the memory use, so ℓ_{mel} and ℓ_{cstft} are only computed on a uniformly sampled subset of the present slots (App. E).

4.3.2 ADDITIONAL TERMS

Due to the sparsity and skewness of our stem distribution, and considering our main train objective, the less frequent stem bins are more prone to *residual dump (bleeding)* or *silence prediction*. We then attempt to regularize those by including some additional terms.

First, we include a *remix consistency* term to discourage zero prediction. Assuming that a rare stem, e.g. sitar, converged to predicting silence; then, including a consistency term for the remix over present stems gives these rare slots another presence in the loss while excluding them when they are absent, penalizing zero prediction. Specifically, we denote the remix $\hat{\omega} = \sum_{i \in \mathcal{P}} \hat{\omega}_{s_i}$, and include the remix consistency term (using α_L^{remix} for each loss L):

$$\mathcal{L}_{\text{remix}} = \ell_1(\{(\hat{\omega}, \omega)\}) + \ell_{\text{mel}}(\{(\hat{\omega}, \omega)\}) + \ell_{\text{cstft}}(\{(\hat{\omega}, \omega)\}) \tag{6}$$

Second, to decrease leakage of residual error into less frequent stems (bleeding), we add another regularization over the absent terms. As an absent slot has no reference, only energy it should not emit, we regularize it with a leakage ratio in the mel domain,

$$\mathcal{L}_{\text{absent}} = \frac{\alpha_{\text{absent}}}{|\mathcal{A}|} \sum_{j \in \mathcal{A}} \frac{\|P_{n_0}(\hat{\omega}_{s_j})\|_1}{\|P_{n_0}(\omega)\|_1}, \tag{7}$$

which equals one when the slot emits the energy of the whole mixture and vanishes as it goes silent.

Last, to support our secondary objective to yield an approximately additive latent representation (Sec. 4.1), we include an additivity term where we define $\hat{\omega}_\Sigma = \frac{1}{|\mathcal{P}|} \sum_{i \in \mathcal{P}} \text{Dec}_i(\sum_{j \in \mathcal{P}} z_{s_j})$, and denote the additivity term (using α_L^{add} for each loss L)

$$\mathcal{L}_{\text{add}} = \ell_1(\{(\hat{\omega}_\Sigma, \omega)\}) + \ell_{\text{mel}}(\{(\hat{\omega}_\Sigma, \omega)\}) + \ell_{\text{cstft}}(\{(\hat{\omega}_\Sigma, \omega)\}). \tag{8}$$

4.3.3 OVERALL OBJECTIVE AND REFINEMENT FINE-TUNING

Eq. 9 presents the overall training objective, in which the optional additivity term is multiplied by an indicator $\mathbb{I}$.

$$\mathcal{L}_{\text{overall}} = \mathcal{L}_{\text{recon}} + \mathcal{L}_{\text{remix}} + \mathcal{L}_{\text{absent}} + \mathbb{I} \cdot \mathcal{L}_{\text{add}}. \tag{9}$$

In practice, we use Défossez et al. (2023)'s gradient balancer module, so the actual weighted sum is being done in a relative manner w.r.t the normalized gradients rather than summing the actual terms in the reported loss (App. E.1).

Following model training we further explore a secondary stage to further mitigate leakage and artifacts by including Défossez et al. (2023)'s discriminator based objectives that include adversarial training and feature matching losses. We include these terms in conjunction with the terms in Eq. 9 and fine-tune the our stage-1 model using single-label examples for additional 50k steps.

5 RESULTS

5.1 EXPERIMENTAL SETUP

We evaluate DeFormer on an internal held-out set and a curated subset of MoisesDB mapped to our taxonomy using song metadata. Since the taxonomies differ (e.g., bowed contrabass is *strings* in ours but *bass* in MoisesDB), we sample tracks by stem activity and randomly select four-second segments until obtaining > 50 candidates per stem. We manually verify and correct their labels where needed (App. I); the subset was fixed independently and prior to model evaluation. Our held-out set contains 2,640 four-second segments, compared with 304 for MoisesDB. As MoisesDB contains no sitar, its mean covers 11 stems. We will release the segment list and stem mapping with the code [1] (App. J).

We report chunk-level cSDR using `museval` v0.4.1,[2] which scores one-second windows and reports their median. Scores are in decibels, $10 \log_{10}(\cdot)$, comparing reference energy to residual energy, with 0 dB indicating equality. We compare against HTDemucs (Rouard et al., 2023), SCNet (Tong et al., 2024), BS-RoFormer (Lu et al., 2024), and Banquet (Watcharasupat & Lerch,

[1] The segment list and stem map are a JSON file containing relative paths, start times, and assigned labels.

[2] https://github.com/sigsep/sigsep-mus-eval

Table 1: **Separation over 12 slots** measuring cSDR (dB), over *re-trained* baselines. Every score is a median over the items of the evaluation set, with *All* the mean over the slots that set can score, and *no-VDB* the same mean excluding bass, drums and vocals. Per-stem intervals in Tab. 6 (App. F.1).

	Method	All	no-VDB	Bass	Drum	Vocl	Keys	SFX	EGtr	Pian	AGtr	Strg	Wind	PPrc	Sitr
MoisesDB	Banquet	2.95 [2.68, 3.39]	0.60	11.17	9.84	6.64	0.28	0.00	1.28	0.90	1.35	0.37	0.64	0.00	–
	HTDemucs	2.10 [1.98, 2.26]	0.19	8.88	7.91	4.76	0.30	0.00	0.43	0.17	0.57	0.02	0.02	0.01	–
	SCNet L	3.62 [3.19, 4.26]	1.03	12.20	11.35	8.07	0.66	0.13	1.83	1.39	2.14	0.95	1.09	0.02	–
	BS-RoFormer	3.82 [3.34, 4.86]	1.52	11.35	10.47	8.04	**1.39**	0.12	2.75	1.95	2.28	1.00	2.64	0.01	–
	DeFormer	4.40 [3.83, 5.52]	1.74	13.41	11.92	9.15	0.99	**0.33**	2.89	**2.42**	3.38	1.37	2.46	0.08	–
	DeFormer (+ adv FT)	**4.61** [4.02, 5.79]	**1.97**	**13.69**	**12.13**	**9.19**	0.90	0.20	**3.47**	2.39	**3.91**	**1.44**	**3.29**	**0.14**	–
Internal	Banquet	4.55 [4.17, 4.99]	2.44	12.72	12.20	7.69	2.93	0.24	3.72	3.21	4.65	2.35	3.46	**0.29**	1.15
	HTDemucs	2.36 [2.10, 2.61]	0.48	9.66	9.50	4.87	1.16	0.00	0.33	0.36	1.82	0.09	0.57	0.00	0.00
	SCNet L	5.26 [4.81, 5.76]	2.90	14.40	13.85	8.83	3.62	0.19	4.66	3.62	5.91	2.61	4.09	0.03	1.33
	BS-RoFormer	5.53 [5.11, 6.02]	3.38	13.37	13.15	9.34	3.57	0.37	4.98	4.27	5.89	3.47	5.27	0.23	2.40
	DeFormer	6.59 [6.01, 7.25]	4.12	16.12	15.47	10.37	**4.76**	0.42	6.14	5.56	7.37	3.99	5.59	0.10	3.17
	DeFormer (+ adv FT)	**6.98** [6.35, 7.69]	**4.61**	**16.39**	**15.50**	**10.39**	4.65	**0.44**	**7.96**	**5.96**	**7.71**	**4.30**	**6.38**	0.26	**3.80**

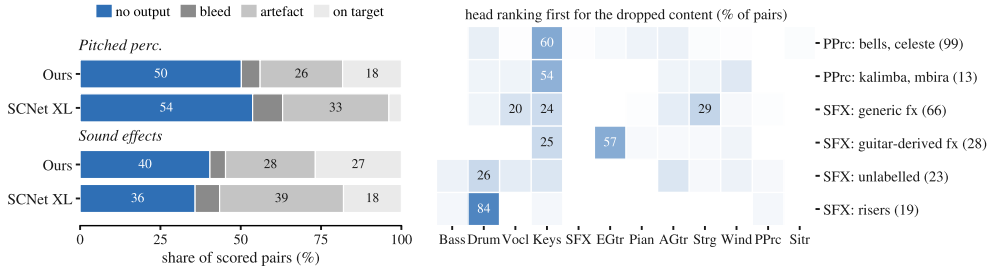

Figure 5: **Failures are dominated by silence and artifact-heavy outputs.** *Left*: fail-case analysis of our two weakest stems. Most failures produce either no output or artifacts; direct bleeding from another present stem is less common. *Right*: where the content of a dropped stem ends up. Each cell is the percentage of that group's content routed to a given head.

2024), all retrained for the expanded stem count on the same corpus and schedule (App. E). BS-RoFormer uses one mask estimator per target, increasing from 132M parameters at 4 stems to 329M at 12. HTDemucs uses its original loss, which performed best in our experiments.

5.2 Objective measures

We start with an analysis of cSDR performance, drawn in Table 1. The measured performance demonstrates a notable advantage for DeFormer across all but PP in our internal held-out set, and all but *keys* and *wind* in MoisesDB; scoring 6.59 dB mean cSDR against 5.53 for the closest baseline. As specified in Sec. 4.3, the last row of each block adds the refinement stage, which lifts the mean to 6.98 and 4.61 dB and gains on most stem types. Sparsity also binds less than expected: *sitar*, our rarest stem type, reaches 3.17 dB, close to the 3.99 dB of *strings*, which is nine times more frequent. We read this as evidence that a timbral profile distinct enough is sufficient for a slot to be learned. To further support that, it appears that the two other most sparse stems, pitched percussion (PP) and sound effects (SFX), fail to show similar performance as sitar and produce near 0 dB cSDR.

To better understand the root cause, we conduct a failure analysis, presented in Fig. 5, sorting every scored (window, slot) pair of the PP and SFX into one of four classes. A pair is *no output* when the estimate's energy falls more than 12 dB below the reference stem's, about 6% of it, so the slot is essentially silent. Otherwise the estimate's STFT magnitude is fitted as a non-negative combination of the stems present in that window, the target's own reference together with the other stems of the mixture, which asks which real sources the output resembles. The pair is then *bleed* when the fit places more weight on the other stems than on the target, *artefact* when the fit residual exceeds half the estimate's total magnitude, so the output matches no stem present, and *on target* otherwise. From Fig. 5(left) we could see that these stems mostly predict silence or suffer from added artifacts. A cell of 0.00 is therefore not partial credit vs the mixture. When observing cases in which they do succeed, they separate as well as anywhere else, at medians of 13.01 and 7.26 dB, so the slot's median counts collapses rather than measuring separation.

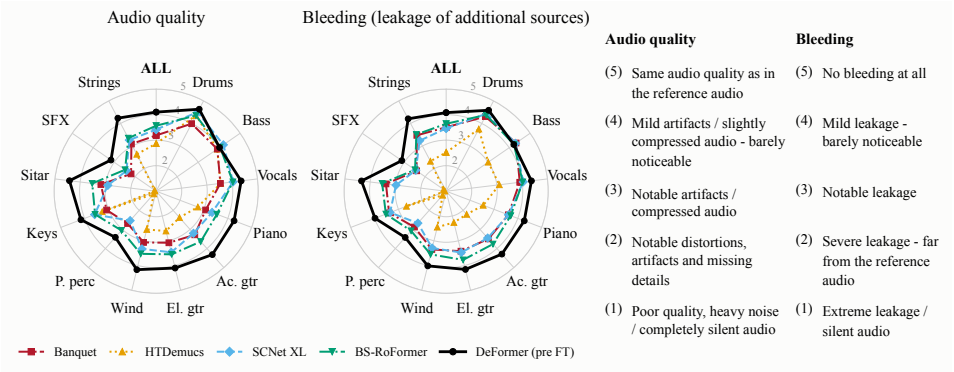

Figure 6: Blind subjective study. A rater was given a reference stem and an estimated stem, and was asked to score *Audio Quality* and *Bleeding* on a scale of 1-5. Calibration point instructions could be found on the right side of the figure. DeFormer scored notably higher on both categories for most stems, suggesting that it has less artifacts and source bleed in the generated stems.

The silence collapse is a routing failure and the energy survives in the other heads (Fig. 5, right): bells are routed predominantly to keys, guitar-derived effects to electric guitar, risers to drums. Both slots are too wide rather than too thin, each collecting content whose salient timbre another slot owns. Where the timbre is distinct the stem is separated successfully: vinyl, scratches and tape hiss reach 6.20 dB, the most buried group we measure. Scarcity alone is not the explanation, since pitched percussion collapses on 51.5 activated hours while sitar does not on 13.9, and the same pattern holds across models, so we hold the target as posed responsible.

5.3 Subjective Study

The cSDR measure rewards waveform agreement, impacted notably by the more energetic components of the spectrum, and could also demonstrate median collapse as happened in SFX and PP. Human listeners would be more sensitive to higher frequency absence than the cSDR measurement, which would be perceptually associated with added artifacts, loss of texture and finer timbral details the objective metric could overlook. For this reason, we conduct a subjective evaluation on 50 segment anchors per stem drawn at random from MoisesDB's subset, with segments added from our held-out set to include *sitar* as well, 600 in total. DeFormer is rated on the pre-refinement checkpoint to match the other models training configuration. We requested annotators to rank *Audio Quality*, and *Bleeding* (adjacent source leakage) on a scale of 1-5 (higher is better), with calibration instructions and examples (Fig. 6, App. G). Ranking was done by 80 people with professional music production backgrounds, each segment was ranked 5 times, so every stem$\times$ model has 250 ratings.

The subjective ranking strengthens the superiority of DeFormer over the observed baselines, further highlighting the spectral modeling advantages our suggested architecture brings. With an average quality score of 4.10 ± 0.76 and bleeding score of 4.08 ± 0.75, DeFormer maintains a ≥ 0.4 score gap from the closest baseline which is *BS-Roformer* with 3.57 ± 0.98 and bleeding score of 3.64 ± 0.97 respectively; also demonstrating the least spread across rankings. We further carried out a qualitative analysis, listening through the estimates of every model on the rated segments ourselves. The cases we find hardest are shared by every system we inspected: two sources whose spectral ranges overlap, playing the same or nearly the same notes in sync, e.g. electric guitar and trumpet, which the taxonomy separates but the spectrum does not. Specific to these "hard" examples, *Banquet* demonstrated less bleed than the other baselines, yet also included notable compression and distortions in the audio. We attribute lower bleed in these cases to the exclusion of the mixture's information preservation from the training objective (single stem extraction); that said DeFormer consistently demonstrated less leakage and better quality than all other baselines over these "hard" cases.

5.4 Ablation Study

This sequential ablation measures the contribution of the complete Fusion and De-Mix module replacements while holding the training scheme fixed. Table 2 starts from SCNet L, the model our design departs from, and adds one component at a time. Neither scale nor the objective explains the

Table 2: Architecture ablation, highlighting the contribution of our suggested components. Starting from SCNet XL, we gain $+0.34$ dB for adding the Fusion module and $+1.30$ dB to also adding the De-Mix module; drawing a large margin from the base SCNet L model.

Configuration	Size (M)	All	no-VDB	Bass	Drum	Vocl	Keys	SFX	EGtr	Pian	AGtr	Strg	Wind	PPrc	Sitr
SCNet L (paper loss)	53	5.12 [4.72, 5.65]	2.80	13.71	13.64	8.93	3.28	0.29	4.45	3.46	5.63	2.68	4.01	0.12	1.26
+ our losses (w.o $\mathcal{L}_{add}$)	53	5.26 [5.07, 5.43]	2.90	14.40	13.85	8.83	3.62	0.19	4.66	3.62	5.91	2.61	4.09	0.03	1.33
scaled up (SCNet XL, own loss)	348	5.20 [4.75, 5.71]	2.80	14.50	14.25	8.44	3.34	0.35	4.70	3.91	5.85	2.34	3.47	0.02	1.25
+ our losses (w.o $\mathcal{L}_{add}$)	348	5.33 [5.12, 5.51]	3.00	14.09	13.76	9.15	3.43	0.20	4.54	3.86	5.71	2.77	4.55	0.04	1.90
+ swap to Fusion	332	5.67 [5.19, 6.28]	3.22	15.15	14.62	9.32	3.81	0.36	5.10	4.22	6.36	2.95	4.38	0.05	1.78
+ swap to De-Mix	156	6.63 [6.08, 7.29]	4.18	16.19	15.34	10.37	**4.81**	0.32	6.16	5.50	7.59	**4.39**	5.75	0.09	3.01
+ add. loss (DeFormer)	156	6.59 [6.34, 6.81]	4.12	16.12	15.47	10.37	4.76	0.42	6.14	5.56	7.37	3.99	5.59	0.10	3.17
+ adv. FT	156	**6.98** [6.74, 7.21]	**4.61**	**16.39**	**15.50**	**10.39**	4.65	**0.44**	**7.96**	**5.96**	**7.71**	4.30	**6.38**	**0.26**	**3.80**
Banquet	25	4.55 [4.40, 4.71]	2.44	12.72	12.20	7.69	2.93	0.24	3.72	3.21	4.65	2.35	3.46	0.29	1.15
+ scaled up	147	4.56 [4.43, 4.72]	2.37	13.01	12.50	7.81	2.83	0.32	3.32	3.24	4.42	2.24	3.04	0.36	1.57

leap over SCNet L. Scaling to 348M parameters buys 0.08 dB in the mean over the 12 slots, 5.12 to 5.20, and our training objective at that size a further 0.13 dB. Our components move the mean considerably further: Fusion alone reaches 5.67 dB, and De-Mix on top of it 6.63 dB at 156M, under half the scaled-up parameter count. This places the gain in the component design as the objective is held fixed across the table, so the components are measured against it rather than confounded with it. Latent approximate additivity is the exception. It leaves the mean where it was, and we keep it for the recombination it buys downstream rather than for separation quality. The refinement stage's gain ("+ adv" row, Sec. 4.3.3) on the internal split averages $+0.39$ dB and shift confidence intervals up. For completeness, the two bottom rows include a scaling experiment for Banquet, scaling it to a similar size as DeFormer, which demonstrate on par performance w.r.t its base configuration.

The four-stem standard is not a target set DeFormer emits directly, and rather than retrain it we "fold" it: vocals, drums and bass each keep their own slot, and the estimated waveforms of the remaining nine are summed into *other*. Table 3 reports the measured cSDR on MUSDB18-HQ against the public baseline checkpoints; Banquet's was not available at the time of writing this manuscript. On that shared setup the folded model sits 1.01 dB above HTDemucs and 0.25 and 0.50 dB below BS-RoFormer and SCNet L, while training the same architecture natively on 4 targets reaches 10.16 dB, inside the intervals of both. We read the top block as context rather than a ranking: those checkpoints are trained on different data mixes and with different training configurations, so the table supports only that an architecture trained on a 4-stem objective lands in the range of established 4-stem systems. The comparison we can make within one data mix is the fold itself, with average -0.35 dB against native 4-stem training, most of it in *other*, the one slot the fold constructs rather than reads off a head (7.16 against 8.25 dB).

Table 3: **4-stem separation on MUSDB18-HQ**, cSDR (dB). At the top, evaluated public checkpoints of our baselines (reference); training data varies. DeFormer (12 stems) is close to the others, despite being trained on a wider target span; 4-stem training show comparable performance.

Model	Bass	Drum	Vocl	Othr	All
HTDemucs	9.82	10.03	8.90	6.45	8.80 [7.93, 9.39]
SCNet L	10.88	11.45	10.73	8.21	10.31 [9.26, 11.01]
BS-RoFormer	9.66	11.68	10.78	8.10	10.06 [9.28, 10.85]
Ours ($K = 4$)	10.51	11.77	10.09	8.25	10.16 [9.35, 10.82]
Ours ($K = 12$)	10.24	11.71	10.13	7.16	9.81 [8.99, 10.34]

6 Conclusion

We introduced DeFormer, a music source separation model for 12 stems, addressing challenges beyond standard four-stem separation. DeFormer separates sources in the bottleneck latent, where target streams interact through joint attention before a shared decoder. Our proposed model components, Fusion and De-Mix, improve mean cSDR by 1.30 dB under a fixed training objective, while a blind listening study gives DeFormer the highest mean ratings for audio quality and freedom from source bleeding.

Sound effects and pitched percussion remain challenging. Distinct sources such as *sitar* separate well even with limited data, whereas spectrally overlapping sources exhibit output collapse, cross-target leakage, or incomplete source assignment across competing stems. These failure modes suggest that broad categories may benefit from partitioning by timbral profile rather than instrument name. We leave this re-annotation and retraining, along with further regularization, to future work.

Ethics Statement

This work relies on two kinds of human participation: the annotation of the training corpus and the listening study of Sec. 5.3. Sourcing and annotation were carried out by external vendors on a platform we provided, against a target profile of music professionals: at least one year of industry experience in music production, a bachelor's degree or higher in music production or composition (with five or more years of industry experience alongside a bachelor's degree), five or more years of industry experience in music composition, or work as a music instructor with more than ten years of experience or in an academic or professional teaching setup. Eligibility against that profile was verified before any annotation took place. Participants supplied judgements about audio material only, and all ratings are reported in aggregate. The audio examples on our samples page are drawn from our held-out set. Code and model checkpoints will be released upon acceptance to support further research.

Reproducibility Statement

The architecture is specified in Sec. 4.2 and App. D, the objective in full in App. H, and the mixture construction, sampler, optimization settings, schedule and gradient balancer in App. E. The evaluation protocol, including the two test sets, the per-slot statistic and the bootstrap procedure behind every interval we report, is given in App. F.1; the corpus annotation protocol is given in App. B and the listening study protocol in App. G. Code and model checkpoints will be released upon acceptance, together with the public MoisesDB segment list, our taxonomy mapping for it and our predictions on it, so that the numbers in Tables 1 and 3 for the public sets can be recomputed end to end.

Use of AI Tools

We used AI assistants as a writing aid, for drafting and revising prose in this paper, and as part of our experiment tracking and reporting tooling. All research claims, experimental designs, measurements and numbers reported here were produced by the authors and verified against the evaluation harness outputs.

References

Estefanía Cano, Gerald Schuller, and Christian Dittmar. Pitch-informed solo and accompaniment separation towards its use in music education applications. *EURASIP Journal on Advances in Signal Processing*, 2014(1):23, 2014. doi: 10.1186/1687-6180-2014-23.

Ke Chen, Xingjian Du, Bilei Zhu, Zejun Ma, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. Zero-shot Audio Source Separation through Query-based Learning from Weakly-labeled Data. In *Proc. AAAI*, 2022.

Woosung Choi, Minseok Kim, Jaehwa Chung, Daewon Lee, and Soonyoung Jung. Investigating U-Nets with various Intermediate Blocks for Spectrogram-based Singing Voice Separation. In *Proc. ISMIR*, 2020.

Alexandre Défossez. Hybrid Spectrogram and Waveform Source Separation. In *Proc. ISMIR Music Demixing Workshop*, 2021.

Alexandre Défossez, Nicolas Usunier, Léon Bottou, and Francis Bach. Music Source Separation in the Waveform Domain, 2019. arXiv:1911.13254.

Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. High Fidelity Neural Audio Compression. *Transactions on Machine Learning Research*, 2023.

Giorgio Fabbro, Stefan Uhlich, Chieh-Hsin Lai, Woosung Choi, Marco Martínez-Ramírez, Weihsiang Liao, Igor Gadelha, Geraldo Ramos, Eddie Hsu, Hugo Rodrigues, Fabian-Robert Stöter, Alexandre Défossez, Yi Luo, Jianwei Yu, Dipam Chakraborty, Sharada Mohanty, Roman Solovyev, Alexander Stempkovskiy, Tatiana Habruseva, Nabarun Goswami, Tatsuya Harada,

Minseok Kim, Jun Hyung Lee, Yuanliang Dong, Xinran Zhang, Jiafeng Liu, and Yuki Mitsufuji. The Sound Demixing Challenge 2023 – Music Demixing Track. *Trans. of the Int. Society for Music Information Retrieval*, 7(1):63–84, 2024.

Juan S. Gómez, Jakob Abeßer, and Estefanía Cano. Jazz solo instrument classification with convolutional neural networks, source separation, and transfer learning. In *Proc. ISMIR*, pp. 577–584, 2018.

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. doi: 10.21105/joss.02154.

Po-Sen Huang, Minje Kim, Mark Hasegawa-Johnson, and Paris Smaragdis. Singing-voice separation from monaural recordings using deep recurrent neural networks. In *Proc. ISMIR*, 2014.

Minseok Kim, Woosung Choi, Jaehwa Chung, Daewon Lee, and Soonyoung Jung. KUIELab-MDX-Net: A Two-Stream Neural Network for Music Demixing. In *Proc. ISMIR Music Demixing Workshop*, 2021.

Jie Hwan Lee, Hyeong-Seok Choi, and Kyogu Lee. Audio query-based music source separation. *arXiv preprint arXiv:1908.06593*, 2019.

Wei-Tsung Lu, Ju-Chiang Wang, Qiuqiang Kong, and Yun-Ning Hung. Music Source Separation with Band-Split RoPE Transformer. In *ICASSP 2024 – IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 481–485, 2024. doi: 10.1109/ICASSP48485.2024.10446843.

Yi Luo and Nima Mesgarani. Conv-TasNet: Surpassing Ideal Time-Frequency Magnitude Masking for Speech Separation. *IEEE/ACM Trans. Audio, Speech, and Language Processing*, 2019.

Yi Luo and Jianwei Yu. Music Source Separation with Band-split RNN. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:1893–1901, 2023. doi: 10.1109/TASLP.2023.3271145.

Ethan Manilow, Prem Seetharaman, and Bryan Pardo. Simultaneous separation and transcription of mixtures with multiple polyphonic and percussive instruments. In *Proc. ICASSP*, pp. 771–775. IEEE, 2020.

Julian D. Parker, Janne Spijkervet, Katerina Kosta, Furkan Yesiler, Boris Kuznetsov, Ju-Chiang Wang, Matt Avent, Jitong Chen, and Duc Le. Stemgen: A music generation model that listens. In *Proc. ICASSP*, pp. 1116–1120. IEEE, 2024.

Igor Pereira, Felipe Araújo, Filip Korzeniowski, and Richard Vogl. MoisesDB: A Dataset for Source Separation Beyond 4-Stems. In *Proc. ISMIR*, 2023.

Zafar Rafii, Antoine Liutkus, Fabian-Robert Stöter, Stylianos Ioannis Mimilakis, and Rachel Bittner. The MUSDB18 corpus for music separation, 2017.

Zafar Rafii, Antoine Liutkus, Fabian-Robert Stöter, Stylianos Ioannis Mimilakis, and Rachel Bittner. MUSDB18-HQ – an uncompressed version of MUSDB18, 2019.

Simon Rouard, Francisco Massa, and Alexandre Défossez. Hybrid Transformers for Music Source Separation. In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023.

Daniel Stoller, Sebastian Ewert, and Simon Dixon. Wave-U-Net: A Multi-Scale Neural Network for End-to-End Audio Source Separation. In *Proc. ISMIR*, 2018.

Fabian-Robert Stöter, Stefan Uhlich, Antoine Liutkus, and Yuki Mitsufuji. Open-Unmix – A Reference Implementation for Music Source Separation. *Journal of Open Source Software*, 4(41): 1667, 2019. doi: 10.21105/joss.01667.

Fabian-Robert Stöter, Antoine Liutkus, and Nobutaka Ito. The 2018 Signal Separation Evaluation Campaign. In *Proc. Latent Variable Analysis and Signal Separation (LVA/ICA)*, 2018.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced Transformer with Rotary Position Embedding. *Neurocomputing*, 568:127063, 2024.

Naoya Takahashi and Yuki Mitsufuji. D3Net: Densely connected multidilated DenseNet for music source separation. *arXiv preprint arXiv:2010.01733*, 2020.

Or Tal, Alon Ziv, Itai Gat, Felix Kreuk, and Yossi Adi. Joint audio and symbolic conditioning for temporally controlled text-to-music generation. *arXiv preprint arXiv:2406.10970*, 2024.

Weinan Tong, Jiaxu Zhu, Jun Chen, Shiyin Kang, Tao Jiang, Yang Li, Zhiyong Wu, and Helen Meng. SCNet: Sparse Compression Network for Music Source Separation. In *Proc. IEEE ICASSP*, 2024.

Stefan Uhlich, Marcello Porcu, Franck Giron, Michael Enenkl, Thomas Kemp, Naoya Takahashi, and Yuki Mitsufuji. Improving music source separation based on deep neural networks through data augmentation and network blending. In *Proc. IEEE ICASSP*, 2017. doi: 10.1109/ICASSP.2017.7952158.

Tuomas Virtanen. Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria. *IEEE Trans. Audio, Speech, and Language Processing*, 15(3):1066–1074, 2007. doi: 10.1109/TASL.2006.885253.

Ju-Chiang Wang, Wei-Tsung Lu, and Minz Won. Mel-Band RoFormer for Music Source Separation. *arXiv preprint arXiv:2310.01809*, 2023.

Karn N. Watcharasupat and Alexander Lerch. A Stem-Agnostic Single-Decoder System for Music Source Separation Beyond Four Stems. In *Proc. ISMIR*, 2024.

Karn N. Watcharasupat, Chih-Wei Wu, Yiwei Ding, Iroro Orife, Aaron J. Hipple, Phillip A. Williams, Scott Kramer, Alexander Lerch, and William Wolcott. A Generalized Bandsplit Neural Network for Cinematic Audio Source Separation. *IEEE Open Journal of Signal Processing*, 5:73–81, 2024.

APPENDIX

A FULL INSTRUMENT TAXONOMY

Table 4: **The full taxonomy and the fold onto the separation targets.** All 98 fine-grained training-taxonomy labels (underscores set as spaces), their 19 canonical families, and the 12 stem group each family folds into. Rows are ordered by model slot, then canonical family. The fold is total: no residual or *other* slot. The annotation interface (Fig. 7) shows its own reference map; headings and examples differ, stem groups do not (App. B).

Stem group	Canonical family	Fine-grained instrument labels
drums	drums	drum kit, drum machine, electric drums, taiko drum, 808, cymbal, gong, cowbell, snare, kick drum, toms, hi hat
	percussion	bongo, conga, djembe, cajon, darbuka, dhol, hand drum, timbale, tambourine, shaker, hand clap, castanet, clave, ethnic percussion, timpani, triangle
bass	bass	electric bass, contrabass (plucked), overdrive electric bass, deep (sub) bass
vocals	vocals	male vocal, female vocal, choir, whistle
piano	piano	piano
ac. guitar	acoustic guitar	acoustic guitar, nylon guitar, resonator guitar, mandolin, ukulele
	plucked strings	banjo, bouzouki, harp, harpsichord, koto, oud, qanun, dulcimer, zither
el. guitar	electric guitar clean	clean electric guitar, pedal steel guitar, slide guitar
	electric guitar distorted	distortion guitar, overdrive guitar, fuzz guitar
wind	brass	trumpet, trombone, horn, tuba, sax
	woodwinds	flute, clarinet, oboe, bassoon, duduk, ney, didgeridoo, ocarina
	free reed	accordion, harmonica, bagpipe
pitched perc.	pitched percussion	bell, marimba, xylophone, vibraphone, glockenspiel, celesta, kalimba, musicbox, steel drum, chime, hang drum
keys	electric piano	electric piano, fm piano, clavinet
	organ	organ
	synth	synthesizer, theremin
sitar	sitar	sitar
sfx	sfx	dj scratch
strings	strings	violin, viola, cello, erhu, pizzicato string, contrabass (bowed)

The 12 separation targets are not the annotators' vocabulary. Annotators work with 98 fine-grained names, and were allowed to add a freeform text field. Each family belongs to exactly one of 19 canonical families, and each family to exactly one of 12 model slots (stem groups). The two maps therefore partition the 98 labels over the 12 slots (Table 4). The fold is total and no slot is trained on a residual: the targets are not 18 families plus a trained *other* slot.

Scoring against MUSDB18-HQ's four-stem references (Table 3) needs a second, coarser fold from our 12 slots. This fold is the identity on the three named sources (*bass*, *drums*, *vocals*) and sums the remaining nine slots into *other*. Each MUSDB estimate sums its assigned slot outputs; nothing is discarded, and the four estimates still sum to the model's own remix. Pitched percussion folds into *other* rather than drums. MUSDB references do the same: only the drum kit itself is the drums stem.

B ANNOTATION PROTOCOL AND INTERFACE

Our data's stem tracks arrive with file names, not labels. A few name themselves ("drums track"), but most stem families must be inferred or annotated. A far larger corpus would absorb the resulting label errors. At 15k songs they land on the rarest families, which the expanded taxonomy exists to recover. We therefore chose clean supervision over noisy supervision on the same 15k songs. The corpus was labeled by 397 annotators with professional music production backgrounds. Each stem was labeled independently by three annotators, who named both an instrument family and specific instruments within it. Figure 7 shows the annotation UI, excluding the questions screen for the two. Reference examples for every family cover the range of sounds an instrument can take.

The disagreement in Sec. 3 is about which families a stem contains. A single-winner majority vote is therefore the wrong rule. Let A_1, A_2, A_3 be the three annotators' family sets. We retain a family

when at least two of them name it,

$$\mathcal{Y} = (A_1 \cap A_2) \cup (A_1 \cap A_3) \cup (A_2 \cap A_3), \tag{10}$$

with $\mathcal{Y}$ the stem's label set. It is multi-valued exactly when annotators split over what is audible, and empty when no two agree. The 94.0% in Sec. 3 is the fraction of stems where Eq. 10 returns a non-empty set. Both consensus rates are counted over families alone; the specific instrument is recorded with each label but does not enter Eq. 10.

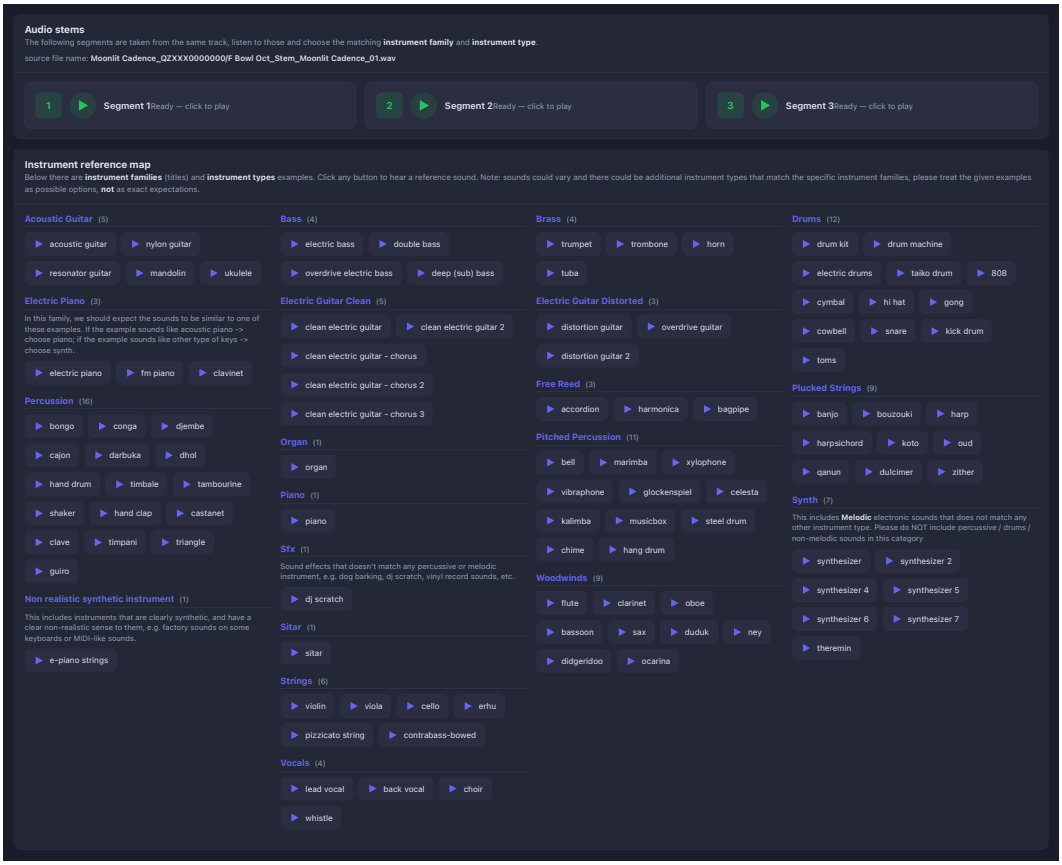

Figure 7: **The stem annotation interface.** An annotator labels one stem with the instrument families it contains. Each example had 3 non-overlapping 5 sec segments randomly drawn from activated regions. Annotators chose an instrument family and a specific instrument type, with freeform text answers allowed. Eq. 10 combines the three annotators' family sets.

C STEM CO-OCCURRENCE

The sampler faces a second constraint, easier to miss than rarity. Co-occurrence between families is strongly asymmetric (Fig. 8). Of songs with electric guitar, 97.3% contain drums; of songs with drums, only 31.5% contain electric guitar. For sitar the gap is starker still: 89.9% against 2.1%. Percentages are computed with respect to *activated* segments. The drums, bass and keys columns are near-saturated for every anchor family. A rare family is therefore never seen in a sparse context: the common families almost always accompany it. Its slot is always learned against a dense background, never against the near-isolated examples that would be easiest to fit.

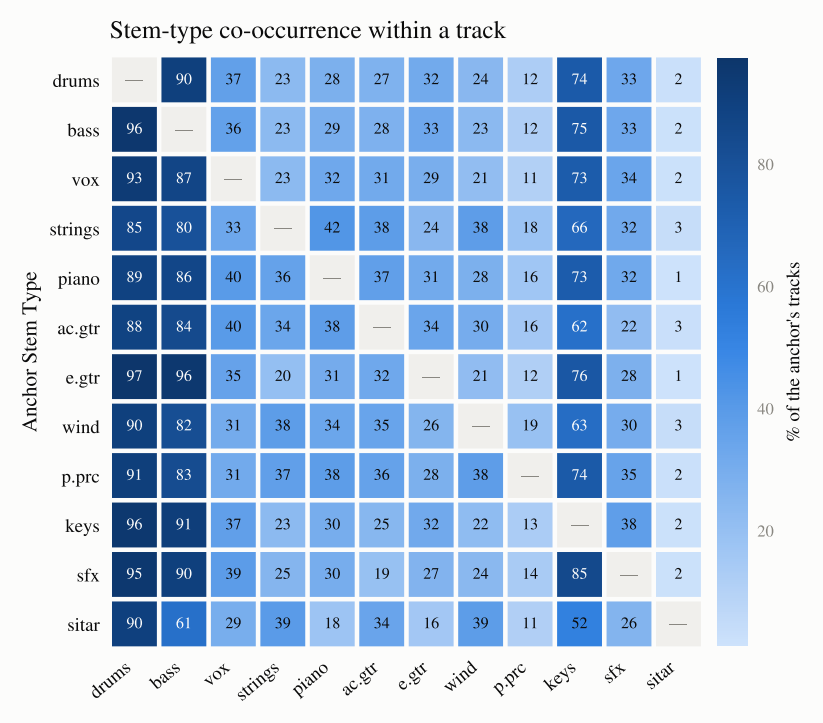

Figure 8: **Stem-type co-occurrence within a track.** Each cell gives $P(\text{column} \mid \text{row})$ over the 15,289 songs: the share of row-type songs that also contain the column type. The anchor's own column is excluded. Rows do not sum to 100: a song contributes to every column it contains.

D Architecture details and hyper-parameters

D.1 Joint attention block and FIS-RoPE

Each joint attention block (Fig. 4a) attends over m sequences at once while keeping them distinct. Every sequence is normalized and projected to its own queries, keys and values. The projections are concatenated along the token axis, one bidirectional attention runs over them, and the result is split back and fed to per-sequence feed-forward layers. A rotary embedding (Su et al., 2024) carries both positions, splitting the head dimension between the two axes (Fig. 4b). Frames at the same instant in different sequences therefore share their per-sequence position. In De-Mix a linear map expands the deepest encoder scale into K streams, each with its own projections and feed-forward weights.

D.2 SD and SU blocks

The SD (sparse down-sample) and SU (sparse up-sample) blocks of Sec. 4.2 come from SCNet unchanged. Both act on the complex spectrogram, compressing frequency only and never subsampling time. An SD block (Fig. 10) splits its input along frequency into low, mid and high bands by fixed proportions. Each band then gets its own strided convolution: a 2-D convolution confined to the frequency axis in kernel and stride, downsampling frequency at every frame. The rates are deliberately unequal. The low band passes at full resolution and the high band is compressed hardest, on the argument that the top of the spectrum carries the least resolvable structure per bin. Each band next passes through its own stack of residual 1-D convolutions along time, with frequency folded into the batch. That stack is deepest for the band compressed least. The bands are then concatenated along frequency and mixed by one 2-D convolution over both axes. An SU block mirrors this: per-band transposed convolutions of the same frequency-only geometry restore each band to its resolution before the matching SD block. The bands are then trimmed to their original lengths and re-joined.

The SU block closing the decoder is its only per-slot component, and the only stem-dependent parameter downstream of De-Mix. It holds K independent sets of three transposed convolutions, one per band: 265,104 parameters, 0.17% of the model and 0.87% of the decoder. This adds no capacity.

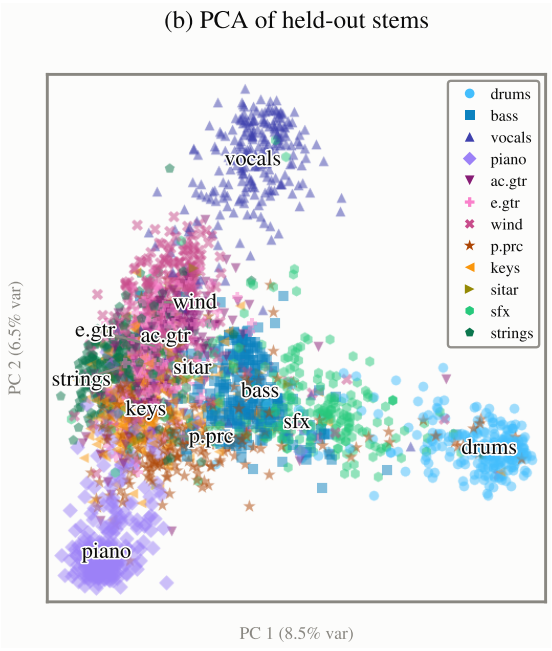

Figure 9: **Held-out stems in the classifier embedding.** First two principal components of the Fig. 1 embedding, per individual held-out stem. Drums and vocals occupy their own regions; the remaining families pile into one central mass. The within-song similarity matrix of Fig. 1 reports the same structure pairwise.

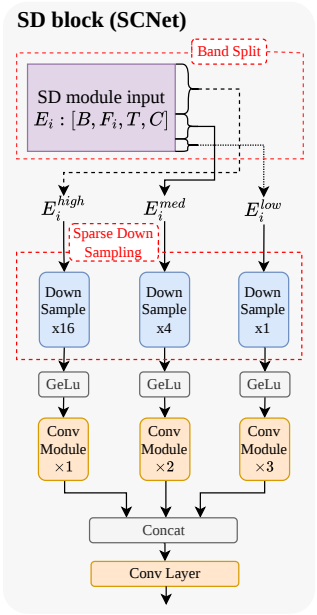

Figure 10: SD block, after SCNet (Tong et al., 2024): bands are downsampled and convolved at their own rates, then concatenated.

The backbone already carries K independent weight sets in its output layer, and K layers of C channels is parameter-identical to one layer of KC channels. Band boundaries stay shared, so the layer learns per-slot emphasis and filtering within a fixed split, not a frequency range. Holding it per slot also breaks the tie between streams at initialization. Under one shared output layer the K streams start at a pairwise cosine of 0.99999 and a per-slot RMS ratio of 1.00: the same signal K times over. A U-Net output is dominated by the encoder skips, which every slot receives identically. An additive per-slot embedding and a per-slot channel affine, both swept over their scale, act upstream of the

skips and leave that cosine at 0.94 or above. The layer is nonetheless affine and reads a shared trunk, so it cannot itself separate: it makes the K streams distinguishable, and De-Mix makes them distinct.

D.3 HYPER-PARAMETERS

Table 5 lists all architecture hyper-parameters for the reported run. Folding the slots onto the batch axis lets the decoder share weights across slots at every stage but the last, whose per-slot output layer is described above. Upstream, De-Mix tells the slots apart through its per-slot projections, feed-forward weights and slot embeddings.

Table 5: Architecture hyper-parameters of DeFormer for the reported run.

Stage	Hyper-parameter	Value
Input	Sample rate f_s	44.1 kHz
	Channels c	2 (stereo)
	Training segment	176,400 samples (4.0 s)
Front-end	FFT size / window	4096 / 4096
	Hop size	1024
	Normalized STFT	yes
Bands	Band fractions (low, mid, high)	0.175 / 0.392 / 0.433
	SD strides per band	1 / 4 / 16
	SD kernels per band	3 / 4 / 16
	Convolution depth per band	3 / 2 / 1
	Compression factor	4
	Convolution kernel	3
Encoder	Stage dims	$[4, 240, 480, 960]$
De-Mix	Joint attention blocks	6
	Slots K	12
	Stream width C_k	896
	Slot embedding std	0.02
	Slot affine (FiLM) std	0.5
Fusion	Joint fusion	at every decoder stage
Decoder	Stage dims	$[960, 480, 240, 4]$
	Weights	shared across slots, all stages but the last
	Final output layer	per slot, affine, 265k params (0.17%)
Total	Parameters	155.79M

E TRAINING CONFIGURATION

This appendix records the training configuration of the reported run.

Task and data. The model predicts $K = 12$ slots in the row order of Table 4: drums, bass, vocals, piano, acoustic guitar, electric guitar, wind, pitched percussion, keys, sitar, sfx, strings. Audio is stereo at 44.1 kHz; training segments are 4.0 s long. The generator has 155,789,552 trainable parameters.

Optimization. We use AdamW with $\beta = (0.9, 0.99)$, $\epsilon = 10^{-8}$, no weight decay and peak learning rate 3×10^{-4}. The schedule warms up over 2,000 steps, holds, then decays as a cosine to zero from step 100,000, and the run reached its full 150,000 optimizer steps. Gradients are clipped to a maximum norm of 1.0. Forward and backward passes run under `bfloat16` autocast with no loss scaler, and no exponential moving average is kept over the weights.

Memory usage of the spectral terms. ℓ_{mel} and ℓ_{cstft} dominate the memory use of the objective, since both are evaluated over $B \times K$ rows. We therefore evaluate ℓ_1 on every present slot but the

two spectral terms on a uniform sample of two slots per example. Each term is normalized over the rows it saw, not over a common count. Present rows are further down-weighted by $\min(1, \tau/\mathcal{L})$ for a per-family threshold τ, bounding an outlier row without discarding it.

Balancer weights. The buckets of Eq. 11 take ℓ_1 at 1.0, ℓ_{mel} at 1.0, ℓ_{cstft} at 2.0, $\mathcal{L}_{\text{absent}}$ at 0.14, $\mathcal{L}_{\text{remix}}$ at 1.0 and $\mathcal{L}_{\text{add}}$ at 0.5. They sum to 5.64, so each term's share of an update is its weight over that total, and ℓ_{cstft} alone takes 35%. Unwanted terms are dropped from the bucket set rather than passed in at zero. The balancer normalizes over the weights it is handed, so a zero bucket would dilute the rest. The reference norm is $N = 1.0$ per batch item, and the moving average over $\|g_b\|$ has decay 0.999.

Outlier trimming. The threshold τ above is a running statistic, not a setting. At each step it takes the 0.95 quantile of the family-normalized per-row losses across all ranks, applied one step stale. The per-family and pooled medians that normalization divides by are tracked by a moving average of decay 0.99, roughly a hundred steps of memory. Trimming is off for the first 500 steps and waits for 200 observations of a family before acting. At most 4,096 rows enter the quantile, and over the reported run τ settled near 3.5.

Sampling. An activated-window sampler assembles every example rather than taking one whole from a track. It draws the anchor uniformly over stem type, so rare families anchor as often as common ones. The window falls uniformly in time inside that anchor's activated regions, so a sparse one-shot part is never sampled as silence. Every other stem of the same track whose activity meets the window joins the mixture, with no minimum overlap and at most 20 stems per example. The stem count is thus what the recording supplies. No same-slot stems are summed into one reference. Slots the track leaves empty are topped up with probability $p_{\text{mix}} = 0.10$. The number of slots to fill is drawn uniformly, and each takes a stem from a track the mixture does not already use. Every added stem is cropped on its own activated regions, so it enters where it is genuinely playing. Activated regions come from a spectral detector over a 2048-point STFT with hop 1024. A frame is active within 35 dB of the stem's own peak and above a floor of -60 dBFS. Regions less than 2.0 s apart are merged; those shorter than 0.25 s are dropped. Two further augmentations apply. A time stretch alters 25% of examples, by a factor drawn from $[0.8, 0.9] \cup [1.1, 1.2]$. A leading or trailing silence of 0.05 to 0.5 s pads 10% of them.

E.1 Gradient balancer

Eq. 9 reads as a weighted sum, but we never form it. Its terms are not commensurable: a leakage ratio and a complex-spectral distance differ by orders of magnitude. Their relative sizes also drift, so fixed weights buy a different trade-off late in training than at the start. The trade-off at stake is the one skew threatens: how much of each update pushes present slots towards their references against how much pushes absent slots towards silence. Under a plain weighted sum, that balance follows the magnitudes the terms happen to take, and hence how many slots each mixture fills. A weight is thus a quantity we set, not one tuned against a loss magnitude, and no term needs normalizing by the energy of its target.

Every term of Eq. 9 is a function of a single tensor y, the stacked per-slot estimates. Whenever $\mathcal{L}_{\text{add}}$ is active, the decoded latent sum $\hat{\omega}_\Sigma$ is appended as one further row. With $g_b = \partial \mathcal{L}_b / \partial y$, the balancer forms

$$G = \sum_{b \in \mathcal{B}} N \frac{g_b}{\text{EMA}(\|g_b\|)} \frac{\alpha_b}{\sum_{b' \in \mathcal{B}} \alpha_{b'}} \tag{11}$$

and backpropagates G into the model parameters, where EMA is an exponential moving average over updates and N a reference norm. The balancer is active in the reported run. Each term is therefore stripped of its own scale and reinjected at the ratio $\alpha_b / \sum_{b'} \alpha_{b'}$, its share of the gradient reaching the model, whatever scale $\mathcal{L}_b$ takes. A share of the update is not a multiplier on a loss value, so no scalar total is meaningful: we report the individual terms, not a summed objective.

F ADDITIONAL RESULTS

F.1 CONFIDENCE INTERVALS

To save width Table 1 carries a song-clustered interval on *All* only; Table 6 reports descriptive window-resampled intervals for every slot. Each bootstrap uses 4,000 replicates at a fixed seed and the 2.5th and 97.5th percentiles. For the per-slot intervals in Table 6, we draw windows with replacement and recompute each slot median and the *All* that follows from them. The resampling unit is the window rather than the (window, slot) pair, so a window enters or leaves a replicate with all of its slots at once, and the slot intervals along a row therefore describe the same resampled sets and can be read against each other. That unit treats windows as independent, which they are not: each set draws roughly twelve windows per song, and windows of one song share instrumentation, mix and difficulty. For the primary intervals we therefore use the song as the unit, so that a drawn song brings all of its windows with it.

Table 6: **Bootstrap confidence intervals on cSDR.** Point estimate and 95% interval per slot for the methods of Table 1, with the best point estimate in each row in bold. Slot names are abbreviated as in Table 1. *All* is the mean over slot medians: eleven slots on MoisesDB, which contributes no sitar material, and all 12 on the internal set.

	Banquet (25M)	HTDemucs (42M)	SCNet L (53M)	BS-RFrmr (329M)	Ours (156M)	Ours (+ adv FT)
MoisesDB						
All	2.95 [2.83, 3.09]	2.10 [2.04, 2.16]	3.62 [3.46, 3.80]	3.82 [3.61, 4.09]	4.40 [4.19, 4.74]	**4.61** [4.37, 4.97]
no-VDB	0.60 [0.51, 0.75]	0.19 [0.15, 0.24]	1.03 [0.87, 1.19]	1.52 [1.29, 1.84]	1.74 [1.53, 2.11]	**1.97** [1.72, 2.41]
Bass	11.17 [10.56, 11.56]	8.88 [8.62, 9.32]	12.20 [11.73, 12.89]	11.35 [10.78, 11.69]	13.41 [12.81, 14.13]	**13.69** [13.11, 14.20]
Drum	9.84 [9.36, 10.24]	7.91 [7.58, 8.25]	11.35 [10.83, 11.82]	10.47 [10.14, 10.97]	11.92 [11.34, 12.43]	**12.13** [11.50, 12.55]
Vocl	6.64 [6.39, 6.98]	4.76 [4.32, 5.05]	8.07 [7.67, 8.49]	8.04 [7.68, 8.44]	9.15 [8.75, 9.51]	**9.19** [8.81, 9.58]
Keys	0.28 [0.16, 0.45]	0.30 [0.19, 0.37]	0.66 [0.34, 0.94]	**1.39** [1.14, 1.75]	0.99 [0.70, 1.38]	0.90 [0.61, 1.41]
SFX	0.00 [0.00, 0.02]	0.00 [0.00, 0.00]	0.13 [0.06, 0.38]	0.12 [0.05, 0.35]	**0.33** [0.08, 0.77]	0.20 [0.02, 0.85]
EGtr	1.28 [1.05, 1.82]	0.43 [0.26, 0.62]	1.83 [1.61, 2.14]	2.75 [2.10, 3.33]	2.89 [2.42, 3.57]	**3.47** [2.86, 4.25]
Pian	0.90 [0.58, 1.05]	0.17 [0.06, 0.38]	1.39 [1.00, 1.70]	1.95 [1.60, 2.28]	**2.42** [1.93, 3.00]	2.39 [1.95, 3.02]
AGtr	1.35 [1.13, 1.76]	0.57 [0.39, 0.69]	2.14 [1.86, 2.42]	2.28 [1.67, 2.70]	3.38 [3.00, 3.82]	**3.91** [3.36, 4.45]
Strg	0.37 [0.14, 0.63]	0.02 [0.01, 0.02]	0.95 [0.56, 1.19]	1.00 [0.57, 1.72]	1.37 [0.86, 1.89]	**1.44** [0.88, 1.81]
Wind	0.64 [0.19, 1.27]	0.02 [0.01, 0.18]	1.09 [0.30, 2.05]	2.64 [1.43, 4.20]	2.46 [1.20, 4.38]	**3.29** [1.98, 4.66]
PPrc	0.00 [0.00, 0.05]	0.01 [0.00, 0.01]	0.02 [0.01, 0.02]	0.01 [0.00, 0.43]	0.08 [0.03, 1.20]	**0.14** [0.03, 2.26]
Sitr	–	–	–	–	–	–
Internal						
All	4.55 [4.40, 4.71]	2.36 [2.25, 2.46]	5.26 [5.07, 5.43]	5.53 [5.36, 5.71]	6.59 [6.34, 6.81]	**6.98** [6.74, 7.21]
no-VDB	2.44 [2.28, 2.62]	0.48 [0.42, 0.56]	2.90 [2.68, 3.07]	3.38 [3.21, 3.60]	4.12 [3.83, 4.38]	**4.61** [4.32, 4.87]
Bass	12.72 [12.43, 13.14]	9.66 [9.40, 9.97]	14.40 [13.99, 14.78]	13.37 [12.98, 13.79]	16.12 [15.75, 16.49]	**16.39** [15.95, 16.80]
Drum	12.20 [11.79, 12.60]	9.50 [9.16, 9.97]	13.85 [13.49, 14.35]	13.15 [12.73, 13.51]	15.47 [15.04, 15.94]	**15.50** [15.08, 15.92]
Vocl	7.69 [6.93, 8.24]	4.87 [3.64, 5.61]	8.83 [8.06, 9.71]	9.34 [8.42, 9.87]	10.37 [9.50, 11.25]	**10.39** [9.46, 11.40]
Keys	2.93 [2.52, 3.23]	1.16 [0.93, 1.41]	3.62 [3.23, 3.97]	3.57 [3.16, 4.12]	**4.76** [4.33, 5.33]	4.65 [4.24, 5.14]
SFX	0.24 [0.06, 0.45]	0.00 [0.00, 0.00]	0.19 [0.09, 0.37]	0.37 [0.13, 0.88]	0.42 [0.13, 1.06]	**0.44** [0.21, 1.31]
EGtr	3.72 [3.00, 4.54]	0.33 [0.09, 0.57]	4.66 [3.85, 5.30]	4.98 [4.42, 5.61]	6.14 [5.47, 6.74]	**7.96** [6.91, 8.55]
Pian	3.21 [2.86, 3.71]	0.36 [0.12, 0.76]	3.62 [3.10, 4.00]	4.27 [3.91, 4.55]	5.56 [5.05, 6.06]	**5.96** [5.52, 6.55]
AGtr	4.65 [4.13, 5.14]	1.82 [1.61, 2.09]	5.91 [5.43, 6.29]	5.89 [5.52, 6.39]	7.37 [7.06, 8.11]	**7.71** [7.18, 8.41]
Strg	2.35 [1.81, 2.69]	0.09 [0.04, 0.21]	2.61 [2.20, 3.08]	3.47 [2.92, 4.00]	3.99 [3.33, 4.73]	**4.30** [3.76, 5.00]
Wind	3.46 [2.88, 4.16]	0.57 [0.26, 0.84]	4.09 [3.13, 4.71]	5.27 [4.63, 5.94]	5.59 [4.10, 6.45]	**6.38** [5.10, 7.29]
PPrc	**0.29** [0.08, 0.71]	0.00 [0.00, 0.00]	0.03 [0.02, 0.11]	0.23 [0.11, 0.73]	0.10 [0.04, 0.35]	0.26 [0.07, 0.93]
Sitr	1.15 [0.80, 1.79]	0.00 [0.00, 0.00]	1.33 [0.32, 2.37]	2.40 [1.62, 3.15]	3.17 [1.38, 4.35]	**3.80** [2.33, 4.55]

F.2 ADDITIVITY OF THE LATENT SPACE

Is the latent space itself additive, or is the decoder trunk merely close to linear? Sec. 4.3 draws that distinction, and two residuals we log during training speak to it. Both ask how well a sum reproduces the mixture. Both are detached, taken over the present slots and normalized by the mixture's own ℓ_1 norm. The *latent path* sums the slot latents and decodes once, $\|\hat{\omega}_\Sigma - \omega\|_1/\|\omega\|_1$, with $\hat{\omega}_\Sigma$ the mean decoded latent sum of Eq. 8. The *waveform path* decodes each slot and sums afterwards, $\|\sum_{i\in\mathcal{P}} \hat{\omega}_{s_i} - \omega\|_1/\|\omega\|_1$. On held-out material the latent path reaches 0.0088 against 0.136 for the waveform path, and the decoded latent sum matches the mixture at 43.7 dB SI-SDR. The two share the present set $\mathcal{P}$ and the same $|\mathcal{P}|$ output-head biases, differing only in whether the latents are summed before or after the trunk. That order-of-magnitude gap therefore measures one thing: how well a summed latent still decodes as the mixture, the property the recombination of Sec. 4.3 relies on. It does not settle additivity on its own, because even a linear trunk S leaves the cross terms $\mathrm{SU}_i(S(z_{s_j}))$, $i \neq j$, in the latent path.

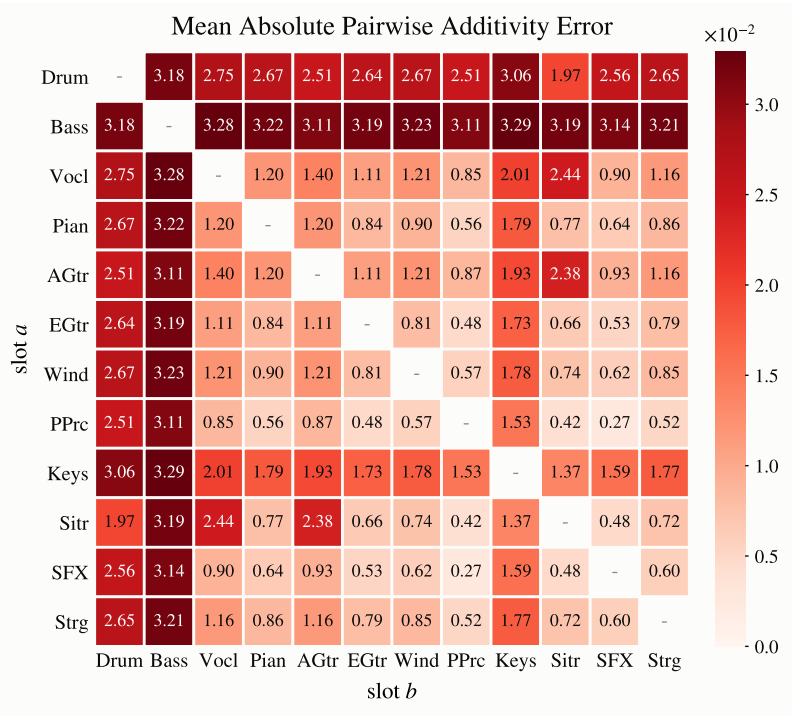

Figure 11: **The latent space is approximately additive, but not pairwise.** Cell (k, j) is the mean of $|\frac{1}{2}\big(\mathrm{Dec}_k(z_{s_k} + z_{s_j}) + \mathrm{Dec}_j(z_{s_k} + z_{s_j})\big) - \mathrm{Dec}_k(z_{s_k}) - \mathrm{Dec}_j(z_{s_j})|$ in units of 10^{-2}, over the held-out segments of the internal split. This is Eq. 8 restricted to the two slots k and j, so the figure measures the objective's own residual on a pair. Exchanging k and j leaves the expression unchanged, so the matrix is symmetric.

A pairwise test settles it, on the same quantity the objective trains. Restricting Eq. 8 to two slots gives $\frac{1}{2}\big(\mathrm{Dec}_k(z_{s_k} + z_{s_j}) + \mathrm{Dec}_j(z_{s_k} + z_{s_j})\big)$, whose target is the sum of the individual estimates, $\mathrm{Dec}_k(z_{s_k}) + \mathrm{Dec}_j(z_{s_j})$. We take the mean ℓ_1 distance between the two, pooled over the 66 slot pairs. It comes to 2.62×10^{-2}, or 37.6% of the mean mixture magnitude (Fig. 11). The space is therefore approximately additive in the sense the objective trains on, yet not in a per-set pairwise additivity manner over an arbitrary pair of slots.

G Listening study protocol

Sec. 5.3 rates five systems on 600 segments, 50 per slot across the 12 slots. A task pairs one estimate with its reference and asks for one property, *Audio Quality* or *Bleeding*, on the five-point scale calibrated beside Fig. 6. Raters never see system identities, nor learn how many models we compare. The order of systems within a segment is shuffled, and tasks go out first come first served, five raters each before the next opens. A rater may therefore judge several systems on one segment, never side by side, and segment assignment is unbalanced by design. We drop every rating the rater flagged for failed playback. That flag is neither uniform nor purely a delivery problem. Most flagged ratings fall on a near-silent render, usually HTDemucs: the model emitted nothing audible, and the rater reported what they heard. Silence is the bottom rung of both scales, so scoring these ratings there is equally defensible. We keep the dropping rule, which credits the baseline rather than our own margin.

Controls and exclusions. We plant controls with known intended ratings: a degraded reference copy (bottom of the quality scale) and a reference carrying a competing stem (bottom of the bleeding scale). A rule fixed in advance drops any rater who placed too few controls near their intended score. No rater triggered it. Every usable rating enters the means, so Sec. 5.3 involves no post hoc rater selection.

H THE OBJECTIVE IN FULL

Notation. Signals ω, ω_{s_i}, $\hat{\omega}_{s_i}$ lie in $\mathbb{R}^{f_s d \times c}$: d seconds of c-channel audio at sample rate f_s. The reconstruction terms, Eqs. **??**, **??** and 4, use two transforms of a signal u. Both come from an n-point Hann-windowed STFT with hop $n/4$. The first, $P_n(u) \in \mathbb{R}^{F \times T_n}$, is the mel-filterbank power spectrogram over $F = 64$ bands at $n_0 = 1024$, averaged over rows. The second, $X_n(u) \in \mathbb{C}^{C \times F_n \times T_n}$, is the corresponding complex STFT.

Grouping multi-slot stems. A multi-label stem cannot be paired slot by slot: the per-slot references would sum to more than the mixture. We therefore group over connected components of the bipartite stem–slot graph, whose edges mark where a stem's label folds into a slot. For a component with stem set A and slot set S_A, we sum estimates and stems separately,

$$\hat{u}_A = \sum_{s \in S_A} \hat{\omega}_s, \qquad u_A = \sum_{k \in A} \omega_k, \tag{12}$$

and evaluate every term of Eq. 9 on the resulting pair. Whole components are needed: two singly-labeled stems folding into the same slot must also be summed into one reference. Unreached slots stay absent; with one label per stem, the construction reduces to the ungrouped case. Multi-slot stems are rare. In the training view, 5.9% of stems carry more than one of the 19 families and 1.06% more than one of the 12 slots. Grouping coarsens supervision within a component, which argues for keeping them rare rather than for a different loss.

I HOW THE CURATED SUBSET OF MOISESDB WAS SELECTED.

We sorted MoisesDB tracks by decreasing number of activated windows and cycled across anchor types, sampling up to ten anchors per type before advancing. For each selected track, we randomly sampled at least four segments containing the anchor and manually verified or corrected their labels against our taxonomy. We continued until collecting at least 50 segments per represented stem type. Selection used no model outputs or separation scores, and the resulting list was frozen before evaluation.

J CLARIFICATION REGARDING MOISESDB'S SEGMENT LIST

We will be sharing a .json file alongside the code for this work. The json will include relative paths to the files we've used for evalution, a list of start time in seconds for the 4 seconds segments, and an assigned *stem label* for each.