FROM TASK AFFINITY TO SHARED COMPUTATION: EFFICIENT SPATIAL AUDIO ENCODING

Anonymous authors
Paper under double-blind review

ABSTRACT

Spatial audio encoders jointly predict sound events, source distances, and directions of arrival. However, assigning a separate processing branch to each task repeats computation even when transformer parameters are shared. We introduce Efficient SpatialAST (ESpAST), a family of encoders guided by layer-wise task-affinity and cross-branch interaction analyses. Representational analysis reveals that sound event detection and distance prediction are most similar in early layers, whereas sound event detection and direction estimation remain comparatively dissimilar. We designed lightweight cross-branch mixers and observed sustained late-layer exchange between branches assigned distance and direction prediction. Together with greater late-layer representational overlap, these learned interactions motivate branch merging, yielding configurations that reduce computation while improving average multi-task performance. On SpatialSoundQA, ESpAST-1.5B$^{\mathcal{M}}_{\mathrm{DOA}}$ achieves a 7.89% average relative improvement across three tasks over a three-branch baseline model, while reducing encoder multiply–accumulate operations by 52.2%. Evaluation with large language models further shows that reduced-computation encoders retain competitive spatial audio question-answering performance. Our results demonstrate how depth-dependent task relationships guide the allocation of shared and task-specific computation, improving the accuracy–computation trade-off of spatial audio encoders.

1 INTRODUCTION

Spatial audio understanding requires recognizing sound events and estimating the direction and distance of their sources. These complementary objectives—sound event detection (**SED**), distance prediction (**DIS**), and direction-of-arrival estimation (**DOA**)—also provide the perceptual information needed for reasoning about acoustic scenes. BAT (Zheng et al., 2024) connects these capabilities to LLM-based spatial reasoning through its Spatial-AST audio encoder, which jointly predicts semantic and spatial information of source audio. DSpAST (Wilkinghoff & Tan, 2026) enhanced Spatial-AST with additional spatial features and task-specific combinations of input features, producing a separate processing branch for each task. Although these branches share transformer parameters, each still requires its own transformer computation. In contrast, our single-branch experiments retain competitive distance prediction but show degraded DOA estimation. These contrasting outcomes motivate a finer-grained approach: *which tasks should share or retain separate processing branches, and at what depths can those branches be combined?*

Prior multi-task learning studies have used task affinity to guide architectural sharing (Vandenhende et al., 2020; Gurulingan et al., 2023) and cross-branch connections to exchange task-specific representations (Misra et al., 2016; Cao et al., 2021). Building on these ideas, we introduce **Efficient SpatialAST (ESpAST)**, a family of spatial audio encoders guided by layer-wise task-affinity and cross-branch interaction analyses. Rather than fixing the same task separation throughout the encoder, we use depth-dependent task relationships to guide both task grouping and the consolidation of later computation. Throughout this investigation, transformer parameters remain shared across branches, allowing us to focus on the organization of their computation.

We first adopt FAMO (Liu et al., 2023) for dynamic task-loss weighting to obtain a stronger and more stable model for architectural comparisons. We then analyze branch representations using centered kernel alignment (Kornblith et al., 2019) and orthogonal Procrustes similarity (Ding et al., 2021).

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.

Despite predicting different types of outputs, SED and DIS exhibit the strongest representational similarity in early layers, whereas SED and DOA remain comparatively dissimilar. DIS–DOA similarity, in contrast, increases in middle-to-late layers. These observations motivate jointly processing SED and DIS while retaining a separate DOA branch. We introduce lightweight bottleneck mixers that enable cross-branch communication and allow us to analyze learned residual contributions. In the three-branch model, DOA-to-DIS residual contributions increase with depth. In the two-branch models that assign DIS and DOA to different branches, the cross-branch residual contributions persist into later layers. Furthermore, these models also exhibit higher inter-branch representational similarity in the later layers than its mixer-free counterparts. Together with the DIS–DOA representational overlap observed in later layers of the three-branch model, these findings motivate replacing later separate processing with a shared branch to reduce computation cost. Comparisons across all three task groupings and multiple merging depths show that keeping DOA separate before merging yields the highest mean multi-task improvement at every tested branch budget, with and without mixers. Several partially merged configurations also outperform both their unmerged counterparts and the FAMO-trained three-branch model in mean multi-task performance. LLM-based evaluation further shows that reduced-computation configurations retain competitive spatial audio question-answering performance. Together, these results demonstrate how task relationships can guide the allocation of shared and task-specific computation in spatial audio encoders.

Our main contributions are:

- We find the strongest early-layer similarity between SED and DIS, greater DIS–DOA similarity in later layers, and sustained late-layer residual exchange in mixer-enabled branches assigned to DIS and DOA, jointly motivating task grouping and later sharing.
- We develop ESpAST with affinity-guided task grouping, lightweight cross-branch mixers, and late merging within a shared-weight transformer. Keeping DOA separate before merging yields the highest mean multi-task improvement among the tested groupings.
- We quantify the accuracy–computation trade-offs of task grouping, merging depth, and mixers. On SpatialSoundQA, ESpAST-$1.5\text{B}^{\mathcal{M}}_{\text{DOA}}$ achieves 7.89% average relative improvement over the fixed-weight three-branch baseline and 5.83 percentage points over its FAMO-trained counterpart, with 52.2% fewer encoder MACs.

2 Related Work

2.1 Multi-task learning

Multi-task-learning (MTL) considers processing multiple related tasks through shared representations or model parameters (Zhang & Yang, 2022). It is hard to optimize because each task requires the model to encode representations in a different way. One research direction to solve this problem is adjusting task-wise loss weights dynamically. Examples include Auto-λ (Liu et al., 2022), NashMTL (Navon et al., 2022), FAMO (Liu et al., 2023) and GRAPE (Fan et al., 2025). Another research direction is architectural search, which determines where task processing should be shared or kept separate. These methods group tasks with high affinity into shared processing paths while assigning separate paths to tasks with low mutual affinity. BMTN (Vandenhende et al., 2020) and MTSL (Gurulingan et al., 2023) measure affinity between tasks based on representation similarity analysis. DTME-MTL (Jeong & Yoon, 2025) and TGLoRA (Gangwar et al., 2026) modify architectures based on cosine similarity between gradients to identify proper locations for divergence, demonstrating that tree-like processing with a shared trunk and task-specific branches can match single-task-optimized models or multi-task models requiring separate task-wise processing.

2.2 Spatial Audio Modeling

The sound event localization and detection task predicts both event categories and the locations of sound sources with respect to recording device (Zhu et al., 2025). Models have improved alongside advances in architectures such as RNNs, CNNs, and transformers (Adavanne et al., 2019; Shimada et al., 2022; Krause et al., 2024). BAT (Zheng et al., 2024) introduced Spatial-AST, a transformer-based audio encoder for spatial audio reasoning LLM. The encoder has been adopted in other LLMs such as OWL (Biswas et al., 2026), NaVLA (Fan et al., 2026) and STAudio-LM (Hyun-Bin et al.,

2026). Prior spatial audio models have explored task-specific processing, cross-task feature exchange, and task-loss weighting. EINv2 (Cao et al., 2021) reports better SELD performance with cross-stitch-based soft parameter sharing than with no sharing or hard sharing. SELD-Mamba (Mu et al., 2026) extends this design to SED, DOA, and distance-estimation branches using bidirectional Mamba blocks. Dong et al. (2025) trained models with different task-wise loss weights over sound event detection and distance estimation tasks and compared effect of joint prediction in pair-wise manner. DSpAST (Wilkinghoff & Tan, 2026) improves Spatial-AST by adding more spatial input features and task-specific input feature attention. Although DSpAST shares transformer parameters, its task-specific branches still require repeated transformer computation. We investigate how much of this separate processing is necessary and at which depths it can be consolidated. Guided by layer-wise task-affinity analysis, we study task grouping, cross-branch communication, and merging into shared later computation to improve the accuracy–computation trade-off of spatial audio encoders.

3 Characterizing Task Relationships in Spatial Audio Encoders

3.1 Baseline Architecture and Experiment Setup

We briefly introduce DSpAST, the baseline model of our study. Spatial-AST is a transformer-based audio encoder and uses left and right channel log-mel spectrograms and cosine, sine of their interaural phase difference (IPD) as inputs as spatial features, stacking them as $x \in \mathbb{R}^{B\times C\times T\times F}$ for batch size B, the number of features C , temporal dimension T and frequency dimension F. DSpAST is built over Spatial-AST, using 2 more input features: Generalized Cross-Correlation with Phase Transform (GCC-PHAT) (Knapp & Carter, 1976) and interaural level difference (ILD) among mel-spectrograms. DSpAST also introduced a feature-attention module which produces masks $\mathcal{F}(x)$ for a batch of input samples $x \in \mathbb{R}^{B\times C\times T\times F}$ in Equation 1. The resulting mask is multiplied element-wise with x, and a convolution-based feature fusion layer produces the task-specific audio tokens $\mathcal{A}_t$. Task-wise [CLS] tokens CLS_t are concatenated to $\mathcal{A}_t$ before entering into shared-weight 12 transformer blocks. The model needs to process transformer blocks per task.

$$\mathcal{F}(x) = \text{Sigmoid}(W_1\ \text{ReLU}(W_0\ \text{BN}(\text{AvgPool}(x)) + b_0) + b_1) \in \mathbb{R}^{B\times C\times 1\times 1} \tag{1}$$

3 task-wise [CLS] tokens are used for sound event detection **(SED)**, direction of arrival prediction **(DOA)** and distance prediction **(DIS)**. The task metrics are mAP over 355 classes for SED, error rate at 20° (ER20) and mean angular error (MAE) for DOA, and distance error ratio (DER) for DIS. We additionally introduce mean distance error (MDE), which measures the average difference between the predicted and ground-truth distances in meters. Binary cross entropy loss is used for SED and AdaCos loss (Zhang et al., 2019) is used for distance prediction, and for azimuth, elevation outputs to predict DOA. DSpAST is trained on SpatialSoundQA: Stage 1 is used only to train the SED capability using pretrained AudioMAE (Huang et al., 2022) weights. From the stage 2 the model is trained with losses from spatial predictions. The 3rd stage mainly boosts additional SED performance from balanced class dataset. Other configurations can be found in Appendix B.

During the reproduction of the baseline model DSpAST, we confirmed that prediction performance fluctuates over training epochs - especially for distance accuracy - which is mentioned in their work. Therefore, we tuned hyper-parameters below to stabilize prediction performance of DSpAST. We applied exponential moving average smoothing to the AdaCos scale factor and capped it at 32. We restored normalization layers applied to spatial branch outputs, which were originally included in Spatial-AST. We set patch embedding layers as branch-wise parameters. We applied bfloat16 mixed precision training (Micikevicius et al., 2018) for speedup. We report each evaluation metric averaged over the final 15 epochs of stage 3. This reporting window was chosen based on the stabilization observed when reproducing DSpAST. We begin our study of **E**fficient **Sp**atial**AST** (ESpAST) by measuring statistics from 3-branch 3B and also a single-branch 1B model which arranges tokens to $[\text{CLS}_{\text{SED}}, \text{CLS}_{\text{DIS}}, \text{CLS}_{\text{DOA}}, \mathcal{A}_i]$. We applied an attention mask over CLS tokens so that they can only interact with audio tokens, following APT (Bowman et al., 2023). We denote the fixed-weight three-branch reference as 3B_{base} and its FAMO-trained counterpart as 3B.

3.2 Stabilizing Training with Dynamic Loss Weighting

We initially explored task grouping based on gradient cosine similarity, as in DTME-MTL and TGLoRA. However, our preliminary measurements fluctuated substantially throughout training (Ap-

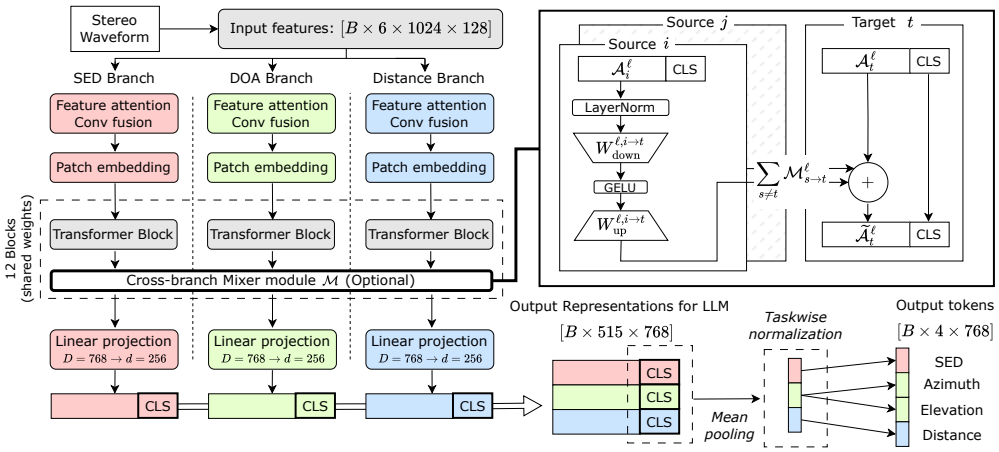

Figure 1: Architecture of ESpAST-3B. Cross-branch Mixer modules are introduced in Section 4.2. Linear projection layers after the last transformer block are task-wise components producing output tokens from their corresponding branches.

pendix A), making the choice of a representative checkpoint or temporal averaging window ambiguous. Before investigating architectural sharing, we therefore adopt FAMO (Liu et al., 2023) as a common optimization procedure across architectures, avoiding separate searches over fixed task-loss weights after each architectural modification. FAMO dynamically adjusts task-wise loss weights (λ) based on changes in log losses to encourage balanced relative task progress. We report model performance with multiple seeds in Table 1. FAMO improved 3B models by scheduling task-wise loss weights as depicted in Appendix C. FAMO also substantially improves the mean and across-seed consistency of angular prediction in the single-branch model while retaining competitive distance prediction. These results suggest that the single-branch model faces challenges in accommodating multiple task objectives within a shared audio-token representation, while dynamic task-loss weighting improves the consistency of angular prediction. Also, 1B models with FAMO show competitive distance prediction against 3B models. This implies independent task-wise branches do not guarantee better performance and joint-prediction can help downstream task performance. Unless otherwise specified, all ESpAST variants are trained with FAMO from stage-2. Following the multi-task learning literature, we report the mean relative improvement of our models over the corresponding baseline seed and tasks, as in Equation 2, where $l_i = 1$ for lower-is-better metrics and 0 otherwise (higher is better). We use mAP, MDE, and MAE for SED, distance, and DOA, respectively. MAE measures angular-error magnitude rather than threshold exceedance, while MDE directly measures absolute distance error in meters.

$$\overline{\Delta}_{\text{MTL}} = \frac{1}{S}\sum_{s=1}^{S}\left[\frac{1}{T}\sum_{i=1}^{T}(-1)^{l_i}(M_{\text{new_model},i,s} - 3\text{B}_{\text{base},i,s})/3\text{B}_{\text{base},i,s}\right], \quad S = 4, T = 3 \tag{2}$$

Table 1: Comparison of fixed and FAMO loss weighting across multiple seeds ($\mathcal{S}$).

		ESpAST-3B					**ESpAST-**1B				
$\mathcal{S}$	λ	**mAP** ↑	**ER20** ↓	**MAE** ↓	**DER** ↓	**MDE** ↓	**mAP** ↑	**ER20** ↓	**MAE** ↓	**DER** ↓	**MDE** ↓
a	Fixed	54.82	19.63	14.24	28.18	0.642	54.02	26.87	20.95	25.19	0.588
	FAMO	55.41	20.55	14.46	25.82	0.593	53.65	20.70	15.77	25.72	0.598
b	Fixed	55.00	21.09	14.61	26.95	0.619	53.59	28.19	21.85	25.48	0.587
	FAMO	55.05	20.43	14.05	26.71	0.614	53.13	20.94	15.98	27.48	0.620
c	Fixed	54.73	21.14	14.95	25.72	0.601	53.12	43.41	29.46	27.28	0.627
	FAMO	54.74	20.03	14.41	24.51	0.581	53.90	23.50	18.33	25.17	0.587
d	Fixed	54.33	20.51	14.89	28.60	0.647	52.95	51.88	37.70	26.17	0.611
	FAMO	54.30	19.56	14.08	28.54	0.645	52.80	20.28	15.50	25.16	0.582

3.3 Layer-Wise Representational Affinity between Spatial Audio Tasks

We aim to reduce the number of independent computation paths, not only to lower computational cost but also to benefit from shared knowledge across tasks. To determine where task-specific processing should remain separate and where tasks can be jointly processed, we analyze how representational similarity evolves as the model processes the input. We measured task-wise affinity across transformer blocks using linear centered kernel alignment (**CKA**) (Kornblith et al., 2019) and a similarity measure based on normalized orthogonal Procrustes distance (**OP**) (Ding et al., 2021). Using the stage 3 training dataset with final epoch's checkpoint, we mean-pool the audio tokens from each task branch (Gurulingan et al., 2023; Chen et al., 2026) to obtain representations X and Y, and center each feature across samples to obtain X_c and Y_c, respectively. CKA uses the squared Frobenius norm of the cross-representation matrix, whereas OP uses its nuclear norm and places less emphasis on dominant singular components (Ding et al., 2021). We use both measures to assess whether task-affinity trends are consistent across these different weightings. For OP, normalizing each nonzero centered representation by its Frobenius norm gives a squared distance in $[0, 2]$. Subtracting half of this distance from one gives a similarity in $[0, 1]$, yielding the OP expression in Equation 3.

$$\mathrm{CKA}(X, Y) = \frac{\|X_c^\top Y_c\|_F^2}{\|X_c^\top X_c\|_F \, \|Y_c^\top Y_c\|_F}, \qquad \mathrm{OP}(X, Y) = \frac{\|X_c^\top Y_c\|_*}{\|X_c\|_F \, \|Y_c\|_F} \tag{3}$$

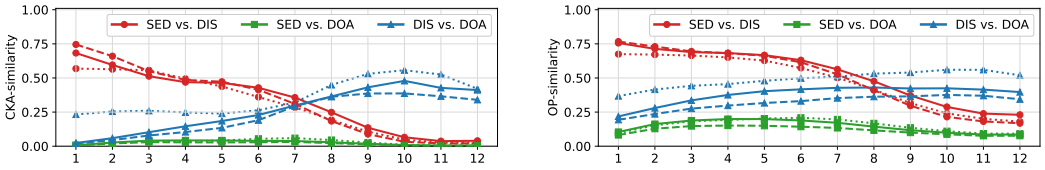

Figure 2: CKA-similarity (left) and OP-similarity (right) over mean-pooled audio tokens across 12 transformer blocks in ESpAST-3B. Similarities between tasks over single-task models were plotted with dash(–) lines. ESpAST-3B trained with Mixer (in Section 4.2) is plotted with dot(..) lines

Because SED does not predict a physical quantity, we expected its representations to be dissimilar from those of DIS and DOA. However, Figure 2 reveals that the (SED, DIS) pair shows the highest similarity among the task pairs in early layers, after which the similarity gradually decreases. Given the competitive DIS accuracy of ESpAST-1B compared with ESpAST-3B, this suggests that jointly processing SED and DIS in early layers may be beneficial. (SED, DOA) pair shows the most disagreement, especially emphasized by **CKA**-similarity. Although the similarity increases in the early-to-middle layers (1–5), it remains substantially lower than that of the other pairs. Given that 1B shows lower DOA performance than the 3B models, with a substantially larger gap without FAMO, this suggests that SED and DOA representations are less compatible within a shared branch, motivating reduced feature sharing between the two tasks. (DIS, DOA) pair shows lower similarity in the early layer. However, it gradually increases through the middle blocks, becomes higher than the (SED, DIS) similarity under both metrics, and then stops increasing at later layers. Note that similar relationship among tasks can be found even if models are trained only with single-task objective for each task t ($\lambda_t = 1, \lambda_{t'} = 0$ for $t' \neq t$) in Figure 2. This indicates that our obtained task-wise similarity is not solely explained by joint optimization under shared transformer blocks.

4 Toward Efficient Spatial Audio Encoders

4.1 Grouping Tasks by Representational Affinity

Motivated by the observed task affinities, we first reduce task-specific computation by assigning two tasks to a shared branch, yielding a two-branch architecture. In ESpAST-2B models, tokens are arranged as [CLS$_{t1}$, CLS$_{t2}$, $\mathcal{A}_i$] and [CLS$_{t3}$, $\mathcal{A}_j$]. We denote each model by the task assigned to branch j (e.g., 2B$_{\mathrm{DOA}}$). Among the mixer-free two-branch configurations, 2B$_{\mathrm{DOA}}$ achieves the highest $\overline{\Delta}_{\mathrm{MTL}}$. It largely preserves mAP and distance performance, although angular performance declines on average. In contrast, 2B$_{\mathrm{SED}}$ primarily degrades distance prediction, whereas 2B$_{\mathrm{DIS}}$, which jointly processes SED and DOA, substantially degrades both mAP and MAE. These results

Table 2: Relative performance of ESpAST configurations against $3B_{base}$, trained with fixed task-wise loss weights. Improvements are computed against the corresponding reference seed and averaged across four seeds. $\mathcal{M}$ denotes mixers, and task subscripts identify the single-task branch before any merging. Absolute metrics are provided in Appendix E, including DSpAST and Spatial-AST.

Model	$\overline{\Delta}_{\text{mAP}} \uparrow$	$\overline{\Delta}_{\text{MAE}} \uparrow$	$\overline{\Delta}_{\text{MDE}} \uparrow$	$\overline{\Delta}_{\text{MTL}} \uparrow \pm$**SD**	**Model**	$\overline{\Delta}_{\text{mAP}} \uparrow$	$\overline{\Delta}_{\text{MAE}} \uparrow$	$\overline{\Delta}_{\text{MDE}} \uparrow$	$\overline{\Delta}_{\text{MTL}} \uparrow \pm$**SD**
3B	0.28%	2.85%	3.04%	$2.06\% \pm 0.4\%$	$3B^{\mathcal{M}}$	0.90%	9.08%	17.88%	$9.28\% \pm 0.5\%$
$2B_{DOA}$	−0.07%	−1.18%	4.09%	$0.95\% \pm 1.2\%$	$2B^{\mathcal{M}}_{DOA}$	0.35%	7.68%	15.08%	$7.70\% \pm 1.4\%$
$2B_{SED}$	0.20%	3.39%	−10.03%	$-2.15\% \pm 2.7\%$	$2B^{\mathcal{M}}_{SED}$	0.38%	4.93%	8.68%	$4.66\% \pm 1.9\%$
$2B_{DIS}$	−5.91%	−10.87%	1.50%	$-5.10\% \pm 6.8\%$	$2B^{\mathcal{M}}_{DIS}$	−2.28%	6.61%	13.48%	$5.94\% \pm 4.2\%$
$1.75B_{DOA}$	−0.18%	6.65%	15.19%	$7.22\% \pm 0.2\%$	$1.75B^{\mathcal{M}}_{DOA}$	0.16%	9.35%	15.96%	$8.49\% \pm 0.8\%$
$1.75B_{SED}$	−1.30%	7.32%	8.02%	$4.68\% \pm 1.7\%$	$1.75B^{\mathcal{M}}_{SED}$	−0.50%	7.74%	9.29%	$5.51\% \pm 2.2\%$
$1.75B_{DIS}$	−2.07%	7.94%	11.99%	$5.96\% \pm 1.4\%$	$1.75B^{\mathcal{M}}_{DIS}$	−1.84%	10.17%	12.72%	$7.02\% \pm 1.6\%$
$1.5B_{DOA}$	−1.13%	4.68%	14.12%	$5.89\% \pm 2.2\%$	$1.5B^{\mathcal{M}}_{DOA}$	−1.05%	9.22%	15.51%	$7.89\% \pm 1.0\%$
$1.5B_{SED}$	−2.70%	5.84%	8.31%	$3.82\% \pm 0.4\%$	$1.5B^{\mathcal{M}}_{SED}$	−1.67%	9.05%	11.66%	$6.35\% \pm 1.6\%$
$1.5B_{DIS}$	−3.31%	1.38%	9.85%	$2.64\% \pm 1.1\%$	$1.5B^{\mathcal{M}}_{DIS}$	−2.73%	6.34%	10.79%	$4.80\% \pm 1.9\%$
$1.25B_{DOA}$	−1.92%	−4.14%	10.87%	$1.60\% \pm 3.4\%$	$1.25B^{\mathcal{M}}_{DOA}$	−1.93%	−1.87%	12.24%	$2.81\% \pm 0.7\%$
$1.25B_{SED}$	−2.33%	−4.75%	9.19%	$0.70\% \pm 1.2\%$	$1.25B^{\mathcal{M}}_{SED}$	−2.69%	−2.18%	10.81%	$1.98\% \pm 2.1\%$
$1.25B_{DIS}$	−3.45%	−3.31%	8.21%	$0.48\% \pm 1.2\%$	$1.25B^{\mathcal{M}}_{DIS}$	−3.10%	−0.53%	8.84%	$1.73\% \pm 1.6\%$
1B	−2.47%	−11.69%	4.82%	$-3.11\% \pm 3.5\%$	$3B_{base}$	0.00%	0.00%	0.00%	$0.00\% \pm 0.00\%$

support the grouping suggested by the early-layer affinity analysis as a favorable overall trade-off among the tested two-branch configurations. We further measure inter-branch representation similarity between two-branch models in Figure 3. $2B_{DOA}$ shows lower inter-branch similarity in early layers, consistent with grouping the most similar pair (SED, DIS) while allocating a separate processing path for the comparatively dissimilar DOA task.

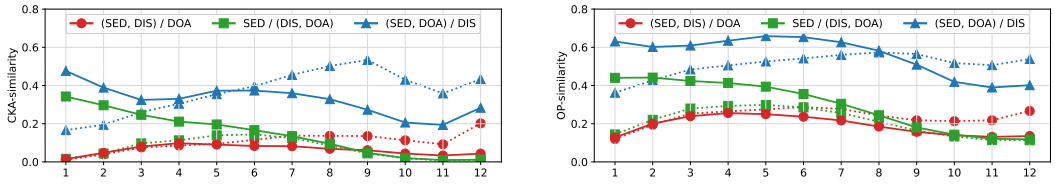

Figure 3: CKA-similarity (left) and OP-similarity (right) over mean-pooled audio tokens measured across 12 transformer blocks in ESpAST-$2B_{DOA}$, ESpAST-$2B_{DIS}$ and ESpAST-$2B_{SED}$, measured from stage-3 training dataset. Dotted lines(..) indicate models trained with Mixer modules.

4.2 Cross-Branch Mixing and Residual Contribution Analysis

In Table 7, ESpAST-1B retains competitive distance prediction despite using a single processing branch, suggesting that a dedicated distance branch is not necessary and that distance prediction can benefit from jointly learned features. Moreover, Figure 2 shows that (DIS, DOA) similarity becomes higher than (SED, DIS) similarity in later layers, suggesting partial representational overlap between the two spatial tasks. Together, these observations motivate us to investigate whether features learned for one task can benefit the other through cross-branch communication. Given the audio-token outputs $\mathcal{A}_s^\ell$ and $\mathcal{A}_t^\ell$ of the source and target branches at the ℓ-th Transformer block, we define a mixer module $\mathcal{M}_{s\to t}^\ell$ that transfers information from $\mathcal{A}_s^\ell$ to $\mathcal{A}_t^\ell$ to produce the updated representation $\widetilde{\mathcal{A}}_t^\ell$, as formulated in Equation 4 and illustrated in Figure 1.

$$\widetilde{\mathcal{A}}_t^\ell = \mathcal{A}_t^\ell + \sum_{s\neq t} \mathcal{M}_{s\to t}^\ell(\mathcal{A}_s^\ell), \quad \mathcal{M}_{s\to t}^\ell(\mathcal{A}_s^\ell) = \text{GELU}\left(\text{LN}(\mathcal{A}_s^\ell)\left(\mathbf{W}_{\text{down}}^{\ell,\,s\to t}\right)^\top\right)\left(\mathbf{W}_{\text{up}}^{\ell,\,s\to t}\right)^\top \quad (4)$$

Our design follows three considerations. First, we limit the parameter and computational overhead by using a bottleneck structure of width $r \ll d$. Second, we avoid restricting cross-branch

transfer to identically indexed channels. Motivated by prior work questioning one-to-one channel correspondence between task representations (Gurulingan et al., 2023), we learn branch-pair specific transformation. Accordingly, each layer-wise mixer consists of LayerNorm (Ba et al., 2016), a down-projection $\mathbf{W}_{\text{down}}^{\ell,s\to t} \in \mathbb{R}^{r\times d}$, GELU (Hendrycks & Gimpel, 2023), and up-projection $\mathbf{W}_{\text{up}}^{\ell,s\to t} \in \mathbb{R}^{d\times r}$. We use $r = 16$ in all experiments. Third, inspired by the zero-initialized updates in LoRA (Hu et al., 2022), we initialize $\mathbf{W}_{\text{up}}^{\ell,s\to t}$ to zero and $\mathbf{W}_{\text{down}}^{\ell,s\to t}$ uniformly at random, so that the cross-branch effect is initially zero. We omit the mixer after the final transformer block because the resulting audio tokens are not used by the prediction heads in ESpAST training and therefore the mixer is not affected by the training loss. The idea of exchanging information between branches over layers was introduced in prior works (Misra et al., 2016; Cao et al., 2021). However, we are not only using it for investigating if representations from one task can help other task's prediction, but also seeking chances for merging branches at later depths to reduce computation. Following Touvron et al. (2021), we quantify learned cross-branch interactions using the residual-update magnitude relative to the target representation for each layer as in Equation 5. A higher normalized residual magnitude indicates a larger additive change relative to the target branch's representation scale.

$$r_\ell^{s\to t} = \left\|\mathcal{M}_{s\to t}^{\ell}(\mathcal{A}_s^{\ell})\right\|_F \Big/ \left(\left\|\mathcal{A}_t^{\ell}\right\|_F + \epsilon\right) \tag{5}$$

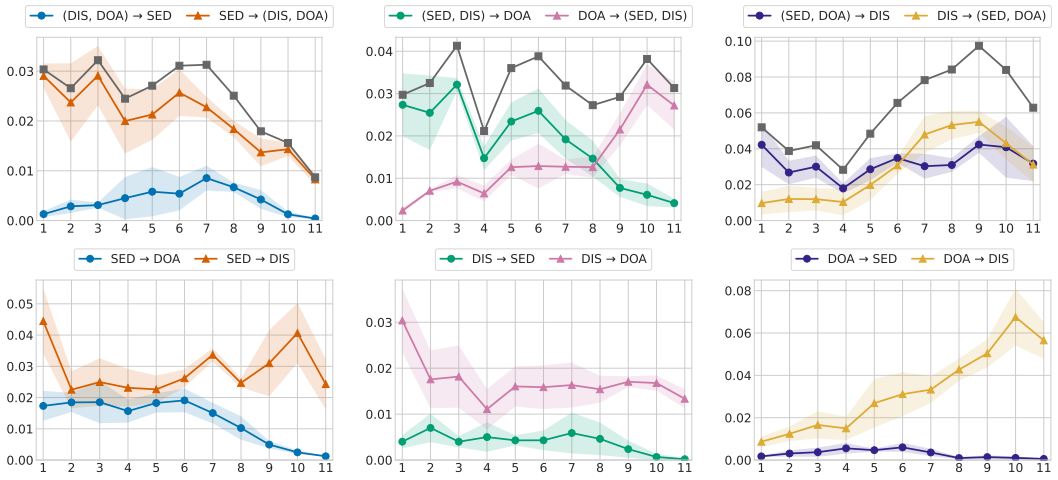

Figure 4: Residual contribution from one branch to another branch per block, measured from $2\text{B}_{\text{SED}}^{\mathcal{M}}$, $2\text{B}_{\text{DOA}}^{\mathcal{M}}$, $2\text{B}_{\text{DIS}}^{\mathcal{M}}$ (upper). Gray lines indicate sum of the two directional ratios. Bottom figures are $r_\ell^{s\to t}$ obtained from $3\text{B}^{\mathcal{M}}$.

With mixer modules, all 2-branch models outperform 3B_{base} in $\overline{\Delta}_{\text{MTL}}$. Both $2\text{B}_{\text{DOA}}^{\mathcal{M}}$ and $2\text{B}_{\text{SED}}^{\mathcal{M}}$ show better accuracy across all tasks, and $2\text{B}_{\text{DIS}}^{\mathcal{M}}$ shows the most difference $\overline{\Delta}_{\text{MTL}}$ against 2B_{DIS}. We also trained a mixer-enabled three-branch model ($3\text{B}^{\mathcal{M}}$) and it achieved the highest $\overline{\Delta}_{\text{MTL}}$ in our experiments. In Figure 4, we visualized trends of $r_\ell^{s\to t}$. Besides comparing task accuracy, $2\text{B}_{\text{SED}}^{\mathcal{M}}$ shows asymmetric amount of residual contribution: $r_\ell^{\text{SED}\to\text{(DIS, DOA)}} > r_\ell^{\text{(DIS, DOA)}\to\text{(SED)}}$. Similar trends also arise on $3\text{B}^{\mathcal{M}}$: $r_\ell^{\text{SED}\to\text{DIS}} > r_\ell^{\text{DIS}\to\text{SED}}$ and $r_\ell^{\text{SED}\to\text{DOA}} > r_\ell^{\text{DOA}\to\text{SED}}$. We further observe that the total residual contribution of $2\text{B}_{\text{SED}}^{\mathcal{M}}$ decreases toward later blocks, whereas non-negligible cross-branch exchange remains in $2\text{B}_{\text{DOA}}^{\mathcal{M}}$ and becomes particularly strong in later layers of $2\text{B}_{\text{DIS}}^{\mathcal{M}}$. Figure 3 also shows that mixer-enabled training changes inter-branch similarity differently by model. Similarity decreases for $2\text{B}_{\text{SED}}^{\mathcal{M}}$, while the best-performing two-branch model, $2\text{B}_{\text{DOA}}^{\mathcal{M}}$, exhibits higher similarity than 2B_{DOA} in later layers. Together with the sustained late-layer cross-branch contributions, this suggests that fully separate computation may become less necessary in later layers. Motivated by this observation, we investigate merging the two branches at later layers.

4.3 Sharing Later Computation through Branch Merging

We further reduce computation by merging two branches i, j at layer m as in Equation 6a, 6b for mixer-enabled models and non-mixer models, respectively.

$$\widetilde{\mathcal{A}}_{\text{merged}} = ((\widetilde{\mathcal{A}}_i^m + \mathcal{M}_{i\to j}^m(\widetilde{\mathcal{A}}_i^m) + \widetilde{\mathcal{A}}_j^m + \mathcal{M}_{j\to i}^m(\widetilde{\mathcal{A}}_j^m)))/2 \quad (6a) \qquad \mathcal{A}_{\text{merged}} = \left(\mathcal{A}_i^m + \mathcal{A}_j^m\right)/2 \quad (6b)$$

We merge the audio tokens while retaining the task-wise [CLS] tokens. 1.25B, 1.5B, and 1.75B denote models whose shared-branch processing begins at 3rd, 6th and 9th transformer block, respectively. We used Multiply-Accumulation-Count (MAC) to report computational costs (Lee et al., 2025) in Table 3. Table 2 shows that keeping DOA separate before merging yields the highest $\overline{\Delta}_{\text{MTL}}$ among the tested task groupings at each branch budget, both with and without mixers. Mixer-free 1.5B and 1.75B configurations also achieve higher mean aggregate performance than their corresponding unmerged two-branch models. With mixers, $1.75\text{B}^{\mathcal{M}}_{\text{DOA}}$ and $1.5\text{B}^{\mathcal{M}}_{\text{DOA}}$ achieve mean relative improvements of 8.49% and 7.89% over 3B_{base}, using only 56.1% and 47.8% of its encoder MACs, respectively. Relative to $2\text{B}^{\mathcal{M}}_{\text{DOA}}$, these configurations retain competitive aggregate performance at lower computational cost. These results support affinity-guided task grouping and later shared computation as a favorable accuracy–computation trade-off.

Table 3: Computational cost of ESpAST configurations. $\mathcal{M}$ denotes models enabled with cross-branch Mixer modules. Relative MAC $\rho\%$ is normalized to the ESpAST-3B model.

	$3\text{B}^{\mathcal{M}}$	3B	$2\text{B}^{\mathcal{M}}$	2B	$1.75\text{B}^{\mathcal{M}}$	1.75B	$1.5\text{B}^{\mathcal{M}}$	1.5B	$1.25\text{B}^{\mathcal{M}}$	1.25B	1B
GMAC ↓	147.4	146.5	98.4	98.1	82.2	82.0	70.1	69.9	57.9	57.8	49.7
ρ (%) ↓	100.6	100.0	67.2	66.9	56.1	56.0	47.8	47.7	39.5	39.5	33.9

4.4 Input-Feature Usage across Branch Configurations

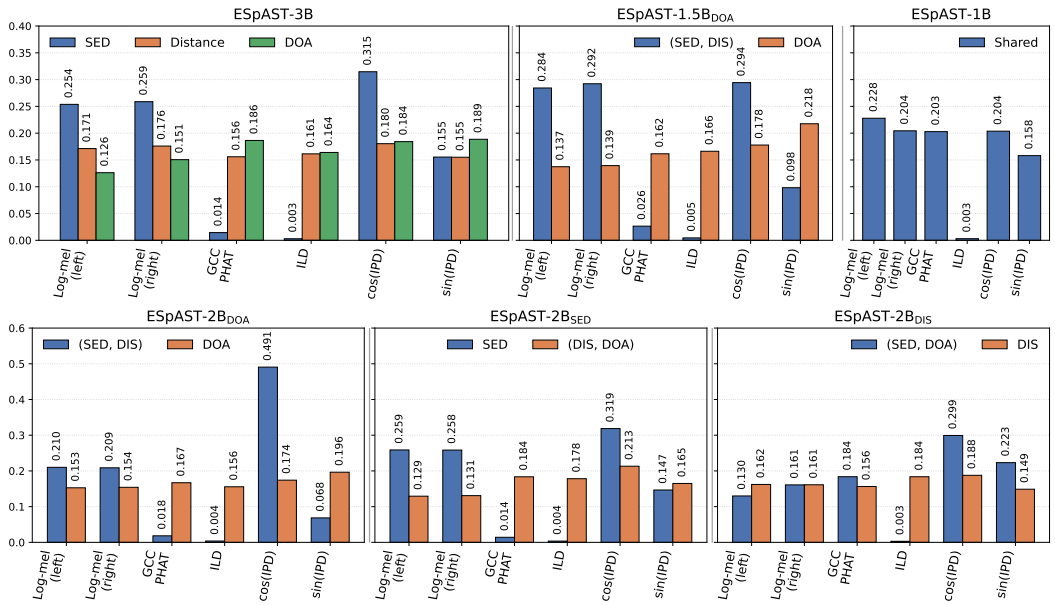

Figure 5: Normalized average input feature usage of the feature attention layer.

In Figure 5, we depicted normalized average weights computed by feature attention (Equation 1) without mixers to examine which input features are emphasized by each branch. 3B and 2B_{DIS} show high ILD usage of distance prediction branch. However, 2B_{DOA} and 1B outperform them while assigning almost no weight to the ILD feature in distance prediction, suggesting that strong distance performance can coexist with low usage of ILD. 2B_{DIS} shows less usage of log-mel spectrograms than other two-branch models, and this may contribute to its lower SED score. Across the configurations shown in the figure, branches assigned DOA consistently allocate appreciable weights to GCC-PHAT, including when DOA is jointly processed with other tasks. In the unmerged two-branch models, this contrasts with the low GCC-PHAT weights of the SED-only and shared SED–DIS branches. However, normalized feature-attention weights do not directly quantify task-specific feature importance. With cross-branch mixers or branch merging, information from either

input branch can contribute to multiple task predictions (Appendix F). We leave the investigation of the relationship between input-feature, branch configuration, and task performance to future work.

5 Spatial Audio Question-Answering Evaluation

We test ESpAST's spatial audio reasoning ability with SpatialSoundQA's LLM training and evaluation dataset, which contains binaural audio–question–answer pairs. We selected branch configurations that keep the DOA processing path separate and use mixer modules, as these models outperform the other settings in the audio-encoder evaluation (e.g., $1.75\text{B}^{\mathcal{M}}_{\text{DOA}}$). Following DSpAST, we used single-stage training with a Q-Former (Li et al., 2023) and MiniLM-L6-v2 (Wang et al., 2020) as the text embedding model. Instead of training Llama-2 7B (Touvron et al., 2023) as backbone LLM, we adopt SmolLM3-3B (Bakouch et al., 2025) as the LLM backbone to reduce computation and memory while retaining strong language modeling capabilities (Appendix D). For DSpAST, we use the audio-encoder checkpoint released in its official GitHub repository. We kept the ESpAST weights frozen and trained only the Q-Former and LLM with LoRA ($r = 16$) to the query and value projections of attention layers, matching similar trainable parameters (3.8M, 8% less than Llama-2 7B with $r = 8$). In Table 4, we found that perception and reasoning do not have identical rankings in the reported means. For example, $3\text{B}^{\mathcal{M}}$ achieves lower distance error than $1.5\text{B}^{\mathcal{M}}_{\text{DOA}}$, whereas the latter achieves higher distance-reasoning accuracy. From $1.25\text{B}^{\mathcal{M}}_{\text{DOA}}$ to $2\text{B}^{\mathcal{M}}_{\text{DOA}}$, our ESpAST models show competitive perception and reasoning scores against DSpAST with lower encoder costs, highlighting the benefits of task-sharing design guided by layer-dependent task relationships.

Table 4: SpatialSoundQA benchmark results with SmolLM3-3B. Types A/B use one source and C/D/E use two. A/C evaluate SED, B/D evaluate direction and distance prediction, and E evaluates spatial reasoning. ESpAST results are averaged over four seeds.

Model	Perception						Reasoning (E)		
	SED ↑		DOA ↑		DER ↓		DOA ↑	Distances ↑	Avg ↑
	A	C	B	D	B	D			
Random	0.61	0.59	12.57	12.41	67.33	67.46	50.00	50.00	50.00
DSpAST	26.15	9.82	79.26	38.67	28.88	48.09	71.14	79.99	75.56
ESpAST-$3\text{B}^{\mathcal{M}}$	27.73	10.58	80.73	39.46	23.70	46.84	71.01	79.74	75.37
ESpAST-$2\text{B}^{\mathcal{M}}_{\text{DOA}}$	27.33	10.57	77.10	37.98	25.33	46.89	69.66	80.69	75.18
ESpAST-$1.75\text{B}^{\mathcal{M}}_{\text{DOA}}$	27.09	10.30	77.60	38.02	25.18	47.32	70.32	80.37	75.34
ESpAST-$1.5\text{B}^{\mathcal{M}}_{\text{DOA}}$	26.66	9.91	80.05	38.30	25.00	48.06	70.55	80.59	75.57
ESpAST-$1.25\text{B}^{\mathcal{M}}_{\text{DOA}}$	26.69	9.56	75.52	36.64	26.90	48.77	70.05	79.74	74.89
ESpAST-1B	25.99	9.21	75.37	37.36	28.43	49.32	70.44	80.19	75.32

6 Conclusion and Discussion

We adopted dynamic task-loss weighting to better exploit each architecture's spatial audio encoding capabilities without separately tuning fixed task-loss weights. Under both multi-task and single-task optimization, SED and DIS exhibited the strongest early-layer representational similarity, motivating joint SED–DIS processing with a separate DOA branch. We introduced lightweight cross-branch mixers and observed asymmetric residual contributions between tasks. Sustained late-layer interactions between branches assigned DIS and DOA further motivated branch merging. Comparisons across task groupings and merging depths support separate early branches followed by shared later computation, with transformer parameters shared throughout. On SpatialSoundQA, ESpAST-$1.5\text{B}^{\mathcal{M}}_{\text{DOA}}$ achieves a 7.89% average relative improvement across tasks over the three-branch baseline while reducing encoder MACs by 52.2%. Exploring combinations with parameter-efficient encoder fine-tuning remains a direction for future work. Overall, our findings demonstrate how depth-dependent task relationships can guide the allocation of shared and task-specific computation, improving the accuracy–computation trade-off of spatial audio encoders. We hope our study will facilitate further research on efficient spatial audio encoders through the lens of multi-task optimization, particularly on when, where, and how task-processing branches should be shared or isolated.

AI USE STATEMENT

In this work, we used generative AI tools to assist with translation and method implementation. We did not use generative AI tools for other tasks requiring disclosure where applicable. Additionally, we used generative AI tools to create or modify scientific figures or images, create or edit software code, summarize or analyze existing literature, identify relevant literature, source or search for information, and edit the research paper to improve readability. We used AI tools to fix grammar, configure visual aspects of figures such as fonts and colors, debug code, and search for and analyze related work relevant to our proposed ideas. We reviewed and verified all AI-assisted work. We take responsibility for the final content of this work, including text, claims, code, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide our implementation in the supplementary material. The SpatialSoundQA dataset and binaural reverberation data used in our experiments are publicly available through the BAT Hugging Face repository. We use the publicly available HuggingFaceTB/SmolLM3-3B and sentence-transformers/all-MiniLM-L6-v2 models, respectively.

REFERENCES

Sharath Adavanne, Archontis Politis, Joonas Nikunen, and Tuomas Virtanen. Sound event localization and detection of overlapping sources using convolutional recurrent neural networks. *IEEE Journal of Selected Topics in Signal Processing*, 13(1):34–48, March 2019. ISSN 1941-0484. doi: 10.1109/jstsp.2018.2885636. URL http://dx.doi.org/10.1109/JSTSP.2018.2885636.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016. URL https://arxiv.org/abs/1607.06450.

Elie Bakouch, Loubna Ben Allal, Anton Lozhkov, Nouamane Tazi, Lewis Tunstall, Carlos Miguel Patiño, Edward Beeching, Aymeric Roucher, Aksel Joonas Reedi, Quentin Gallouédec, Kashif Rasul, Nathan Habib, Clémentine Fourrier, Hynek Kydlicek, Guilherme Penedo, Hugo Larcher, Mathieu Morlon, Vaibhav Srivastav, Joshua Lochner, Xuan-Son Nguyen, Colin Raffel, Leandro von Werra, and Thomas Wolf. SmolLM3: smol, multilingual, long-context reasoner. https://huggingface.co/blog/smollm3, 2025.

Subrata Biswas, Mohammad Nur Hossain Khan, and Bashima Islam. OWL : Geometry-aware spatial reasoning for audio large language models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=zPv46YKv3w.

Benjamin Bowman, Alessandro Achille, Luca Zancato, Matthew Trager, Pramuditha Perera, Giovanni Paolini, and Stefano Soatto. À-la-carte prompt tuning (apt): Combining distinct data via composable prompting. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14984–14993, 2023. doi: 10.1109/CVPR52729.2023.01439.

Yin Cao, Turab Iqbal, Qiuqiang Kong, Fengyan An, Wenwu Wang, and Mark D. Plumbley. An improved event-independent network for polyphonic sound event localization and detection. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 885–889, 2021. doi: 10.1109/ICASSP39728.2021.9413473.

Chuyang Chen, Sivan Ding, Adrian S. Roman, and Juan P. Bello. Probing spatial structure in pretrained audio representations, 2026. URL https://arxiv.org/abs/2606.05544. Accepted to Interspeech 2026.

Frances Ding, Jean-Stanislas Denain, and Jacob Steinhardt. Grounding representation similarity through statistical testing. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=_kwj6V53ZqB.

Yuxuan Dong, Qing Wang, Hengyi Hong, Ya Jiang, and Shi Cheng. An experimental study on joint modeling for sound event localization and detection with source distance estimation. In *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, 2025. doi: 10.1109/ICASSP49660.2025.10887663.

Jugang Fan, Peihao Chen, Changhao Li, Qing Du, Jian Chen, and Mingkui Tan. Navla^2: A vision-language-audio-action model for multimodal instruction navigation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40:18234–18242, 03 2026. doi: 10.1609/aaai.v40i22.38886.

Simin Fan, Maria Ios Glarou, and Martin Jaggi. GRAPE: Optimize data mixture for group robust multi-target adaptive pretraining. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=JRmIvBcnWc.

Neeraj Gangwar, Anshuka Rangi, Rishabh Deshmukh, Holakou Rahmanian, Yesh Dattatreya, and Nickvash Kani. Parameter-efficient multi-task learning via progressive task-specific adaptation. In Emtiyaz Khan, Yingzhen Li, Arno Solin, and Aaditya Ramdas (eds.), *Proceedings of The 29th International Conference on Artificial Intelligence and Statistics*, volume 300 of *Proceedings of Machine Learning Research*, pp. 3916–3924. PMLR, 02–05 May 2026. URL https://proceedings.mlr.press/v300/gangwar26a.html.

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 *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 776–780, 2017. doi: 10.1109/ICASSP.2017.7952261.

Naresh Kumar Gurulingan, Bahram Zonooz, and Elahe Arani. Multi-task structural learning using local task similarity induced neuron creation and removal. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 12205–12223. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/gurulingan23a.html.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus), 2023. URL https://arxiv.org/abs/1606.08415.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Po-Yao Huang, Hu Xu, Juncheng Li, Alexei Baevski, Michael Auli, Wojciech Galuba, Florian Metze, and Christoph Feichtenhofer. Masked autoencoders that listen. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Oh Hyun-Bin, Kazuki Shimada, Yuhta Takida, Kim Sung-Bin, Toshimitsu Uesaka, Takashi Shibuya, Kyeongyoon Lee, Tae-Hyun Oh, and Yuki Mitsufuji. Spatio-temporal audio language modeling for dynamic sound sources, 2026. URL https://arxiv.org/abs/2606.14141. Accepted to EMNLP 2026.

Wooseong Jeong and Kuk-Jin Yoon. Resolving token-space gradient conflicts: Token space manipulation for transformer-based multi-task learning. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2887–2897, 2025. doi: 10.1109/ICCV51701.2025.00277.

C. Knapp and G. Carter. The generalized correlation method for estimation of time delay. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 24(4):320–327, 1976. doi: 10.1109/TASSP.1976.1162830.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3519–3529. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/kornblith19a.html.

Daniel Aleksander Krause, Archontis Politis, and Annamaria Mesaros. Sound event detection and localization with distance estimation. In *2024 32nd European Signal Processing Conference (EUSIPCO)*, pp. 286–290, 2024. doi: 10.23919/EUSIPCO63174.2024.10715220.

DongJae Lee, Jiwan Hur, Jaehyun Choi, Jaemyung Yu, and Junmo Kim. Frequency-aware token reduction for efficient vision transformer. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 34058–34083. Curran Associates, Inc., 2025. doi: 10.52202/085713-1144. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/30e15e5941ae0cdab7ef58cc8d59a4ca-Paper-Conference.pdf.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Bo Liu, Yihao Feng, Peter Stone, and qiang liu. FAMO: Fast adaptive multitask optimization. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=zMeemcUeXL.

Shikun Liu, Stephen James, Andrew Davison, and Edward Johns. Auto-lambda: Disentangling dynamic task relationships. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL https://openreview.net/forum?id=KKeCMim5VN.

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, and Hao Wu. Mixed precision training. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=r1gs9JgRZ.

Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3994–4003, 2016. doi: 10.1109/CVPR.2016.433.

Da Mu, Zhicheng Zhang, Haobo Yue, Zehao Wang, Jin Tang, and Jianqin Yin. Seld-mamba: Selective state space model for sound event localization and detection with source distance estimation. In Huaping Liu and Di Guo (eds.), *Proceedings of 2025 Chinese Intelligent Automation Conference*, pp. 176–183, Singapore, 2026. Springer Nature Singapore. ISBN 978-981-95-4053-2.

Aviv Navon, Aviv Shamsian, Idan Achituve, Haggai Maron, Kenji Kawaguchi, Gal Chechik, and Ethan Fetaya. Multi-task learning as a bargaining game. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 16428–16446. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/navon22a.html.

Kazuki Shimada, Yuichiro Koyama, Shusuke Takahashi, Naoya Takahashi, Emiru Tsunoo, and Yuki Mitsufuji. Multi-accdoa: Localizing and detecting overlapping sounds from the same class with auxiliary duplicating permutation invariant training. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 316–320, 2022. doi: 10.1109/ICASSP43922.2022.9746384.

Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 32–42, 2021. doi: 10.1109/ICCV48922.2021.00010.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi,

Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023. URL https://arxiv.org/abs/2307.09288.

Simon Vandenhende, Stamatios Georgoulis, Bert De Brabandere, and Luc Van Gool. Branched multi-task networks: Deciding what layers to share, 2020. URL https://arxiv.org/abs/1904.02920. Accepted at BMVC 2020.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Kevin Wilkinghoff and Zheng-Hua Tan. Dspast: Disentangled representations for spatial audio reasoning with large language models. In *ICASSP 2026 - 2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 14747–14751, 2026. doi: 10.1109/ICASSP55912.2026.11460926.

Xiao Zhang, Rui Zhao, Yu Qiao, Xiaogang Wang, and Hongsheng Li. Adacos: Adaptively scaling cosine logits for effectively learning deep face representations. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10815–10824, 2019. doi: 10.1109/CVPR.2019.01108.

Yu Zhang and Qiang Yang. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, 34(12):5586–5609, 2022. doi: 10.1109/TKDE.2021.3070203.

Zhisheng Zheng, Puyuan Peng, Ziyang Ma, Xie Chen, Eunsol Choi, and David Harwath. BAT: Learning to reason about spatial sounds with large language models. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=kao5hRX9YA.

Zhiyuan Zhu, Yu Zhang, Wenxiang Guo, Changhao Pan, and Zhou Zhao. ASAudio: A survey of advanced spatial audio research. In Kentaro Inui, Sakriani Sakti, Haofen Wang, Derek F. Wong, Pushpak Bhattacharyya, Biplab Banerjee, Asif Ekbal, Tanmoy Chakraborty, and Dhirendra Pratap Singh (eds.), *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pp. 417–442, Mumbai, India, December 2025. The Asian Federation of Natural Language Processing and The Association for Computational Linguistics. ISBN 979-8-89176-298-5. doi: 10.18653/v1/2025.ijcnlp-long.25. URL https://aclanthology.org/2025.ijcnlp-long.25/.

A Task-Wise Gradient Cosine Similarity

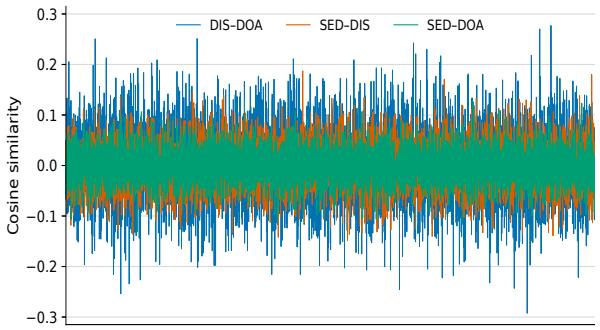

Figure 6: Cosine-similarity of task-wise gradients measured during training ESpAST-3B.

B Training Configuration and Hyperparameters

We kept most of the training hyperparameter used in DSpAST for controlled comparison. Table 5 uses the same setting as DSpAST. For FAMO, we followed its default setting ($\gamma = 0.001, \alpha = 0.025$). We used the same training and evaluation dataset in our experiments following DSpAST.

Table 5: Training hyperparameter in ESpAST.

	Stage 1	**Stage 2**	**Stage 3**
AudioSet split (Gemmeke et al., 2017)	Unbalanced	Unbalanced	Balanced
Weighted sampling	Yes	Yes	No
Dataset usage per epoch	10%	1%	100%
Fixed loss weights $(\lambda_{\text{SED}}, \lambda_{\text{DIS}}, \lambda_{\text{AZI}}, \lambda_{\text{ELE}})$	$(1, 0, 0, 0)$	$(100, 2, 1, 1)$	$(100, 2, 1, 1)$
Number of epochs	100	50	50
Warm-up epochs	10	10	10
Effective batch size	128	128	128
Base learning rate	1×10^{-3}	1×10^{-3}	1×10^{-4}

C Dynamic Task-Wise Loss Weights

FAMO allocates large weights to SED (λ_{SED}) throughout training. The algorithm gives weights to distance prediction (λ_{DIS}) relatively larger than fixed-weighting in the early stage. Also, trends of λ_{Azimuth} and $\lambda_{\text{Elevation}}$ show different sensitivity within direction of arrival prediction task in early epochs.

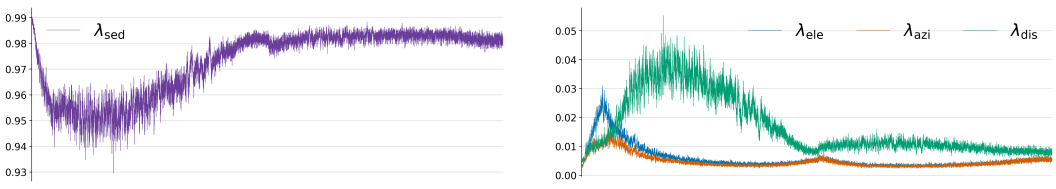

Figure 7: Task-wise loss weights over training steps over the 2nd and the 3rd stage with FAMO.

D LLM BACKBONE COMPARISON

Table 6: Comparison of the base language models. The results are provided as a reference for general language-model capability. HellaSwag, PIQA, WinoGrande, and BoolQ are evaluated in the zero-shot setting. For GSM8K, LLaMA-2 7B and SmolLM3-3B report 8-shot and 5-shot results, respectively.

Model	Params.	HellaSwag ↑	PIQA ↑	WinoGrande ↑	BoolQ ↑	GSM8K ↑
LLaMA-2 7B	7B	77.20	78.80	69.20	77.40	14.60
SmolLM3-3B-Base	3B	76.15	78.89	58.88	78.99	67.63

E ESPAST MODEL PERFORMANCE

Table 7: Performance of ESpAST configurations averaged across four seeds. $\overline{\Delta}_{\text{MTL}}$ is computed against the corresponding seed of 3B_{base}, trained with fixed task-wise loss weights. $\mathcal{M}$ denotes cross-branch mixer modules, and task subscripts identify the single-task branch before any merging. Results for Spatial-AST (Zheng et al., 2024) and DSpAST (Wilkinghoff & Tan, 2026) are taken from their respective papers.

Model	$\overline{\Delta}_{\text{MTL}}$	mAP ↑	ER20 ↓	MAE ↓	DER ↓	MDE ↓	Model	$\overline{\Delta}_{\text{MTL}}$	mAP ↑	ER20 ↓	MAE ↓	DER ↓	MDE ↓
3B	2.06%	54.88	20.14	14.25	26.39	0.608	$3\text{B}^{\mathcal{M}}$	9.28%	55.21	18.94	13.34	20.92	0.515
2B_{DOA}	0.95%	54.68	20.71	14.83	25.52	0.601	$2\text{B}^{\mathcal{M}}_{\text{DOA}}$	7.70%	54.91	19.19	13.54	22.14	0.533
2B_{SED}	-2.15%	54.83	19.72	14.17	31.11	0.690	$2\text{B}^{\mathcal{M}}_{\text{SED}}$	4.66%	54.93	19.58	13.95	24.38	0.573
2B_{DIS}	-5.10%	51.48	21.15	16.29	26.82	0.617	$2\text{B}^{\mathcal{M}}_{\text{DIS}}$	5.94%	53.47	18.43	13.68	22.76	0.543
$1.75\text{B}_{\text{DOA}}$	7.22%	54.62	19.43	13.70	22.00	0.532	$1.75\text{B}^{\mathcal{M}}_{\text{DOA}}$	8.49%	54.81	18.86	13.30	21.68	0.527
$1.75\text{B}_{\text{SED}}$	4.68%	54.01	18.92	13.59	24.53	0.577	$1.75\text{B}^{\mathcal{M}}_{\text{SED}}$	5.51%	54.45	18.83	13.53	24.26	0.569
$1.75\text{B}_{\text{DIS}}$	5.96%	53.59	18.74	13.50	23.36	0.552	$1.75\text{B}^{\mathcal{M}}_{\text{DIS}}$	7.02%	53.71	18.08	13.17	23.10	0.547
1.5B_{DOA}	5.89%	54.10	19.82	13.98	22.21	0.538	$1.5\text{B}^{\mathcal{M}}_{\text{DOA}}$	7.89%	54.14	18.91	13.31	21.73	0.530
1.5B_{SED}	3.82%	53.25	19.00	13.81	24.42	0.575	$1.5\text{B}^{\mathcal{M}}_{\text{SED}}$	6.35%	53.80	18.33	13.33	23.55	0.554
1.5B_{DIS}	2.64%	52.91	19.56	14.46	23.92	0.566	$1.5\text{B}^{\mathcal{M}}_{\text{DIS}}$	4.80%	53.23	18.60	13.74	23.61	0.560
$1.25\text{B}_{\text{DOA}}$	1.60%	53.67	20.76	15.27	23.38	0.559	$1.25\text{B}^{\mathcal{M}}_{\text{DOA}}$	2.81%	53.66	20.61	14.94	22.87	0.550
$1.25\text{B}_{\text{SED}}$	0.70%	53.44	20.49	15.37	24.02	0.570	$1.25\text{B}^{\mathcal{M}}_{\text{SED}}$	1.98%	53.25	20.14	14.98	23.47	0.559
$1.25\text{B}_{\text{DIS}}$	0.48%	52.83	20.25	15.15	24.37	0.575	$1.25\text{B}^{\mathcal{M}}_{\text{DIS}}$	1.73%	53.02	19.82	14.75	24.15	0.572
1B	-3.11%	53.37	21.35	16.39	25.88	0.597	3B_{base}	0.00%	54.72	20.59	14.67	27.36	0.628
Spatial-AST	-	50.03	23.89	17.94	32.54	-	DSpAST	-	54.53	20.28	14.44	28.03	-

F INPUT-FEATURE ATTENTION WEIGHTS

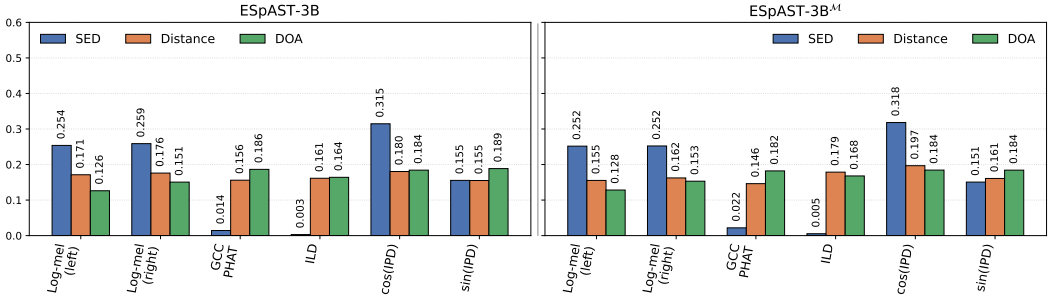

Figure 8: Normalized average input feature usage of the feature attention layer in ESpAST-3B and ESpAST-$3\text{B}^{\mathcal{M}}$.

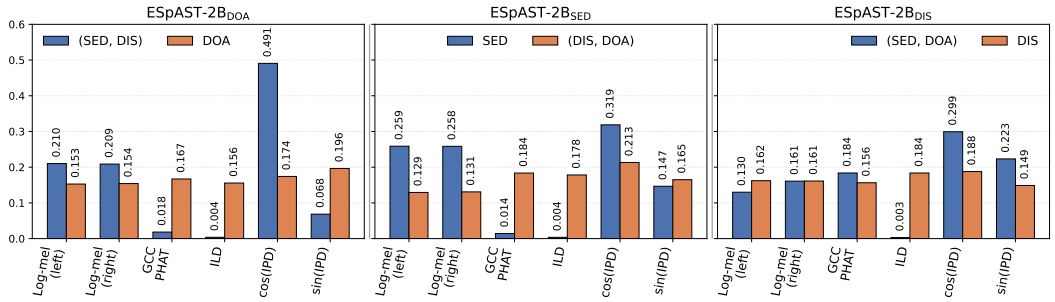

Figure 9: Normalized average input feature usage of the feature attention layer in ESpAST-2B$_{DOA}$, ESpAST-2B$_{SED}$ and ESpAST-2B$_{DIS}$.

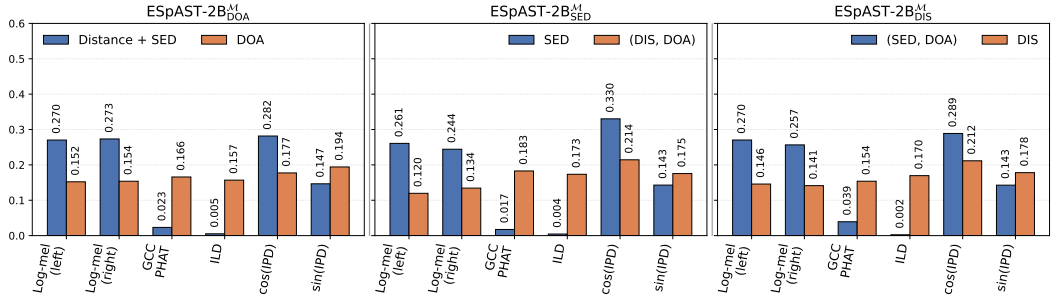

Figure 10: Normalized average input feature usage of the feature attention layer in ESpAST-2B$^{\mathcal{M}}_{DOA}$, ESpAST-2B$^{\mathcal{M}}_{SED}$ and ESpAST-2B$^{\mathcal{M}}_{DIS}$.

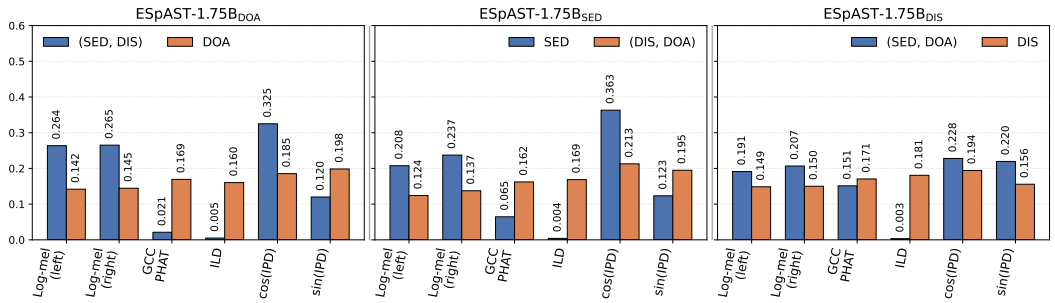

Figure 11: Normalized average input feature usage of the feature attention layer in ESpAST-1.75B$_{DOA}$, ESpAST-1.75B$_{SED}$ and ESpAST-1.75B$_{DIS}$.

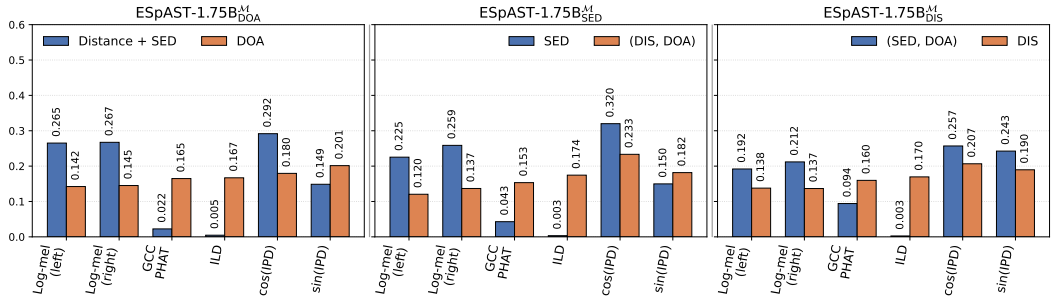

Figure 12: Normalized average input feature usage of the feature attention layer in ESpAST-1.75B$^{\mathcal{M}}_{DOA}$, ESpAST-1.75B$^{\mathcal{M}}_{SED}$ and ESpAST-1.75B$^{\mathcal{M}}_{DIS}$.

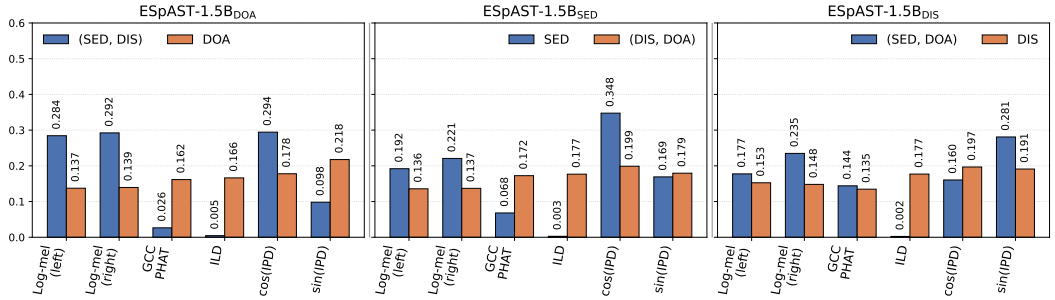

Figure 13: Normalized average input feature usage of the feature attention layer in ESpAST-1.5B$_{DOA}$, ESpAST-1.5B$_{SED}$ and ESpAST-1.5B$_{DIS}$.

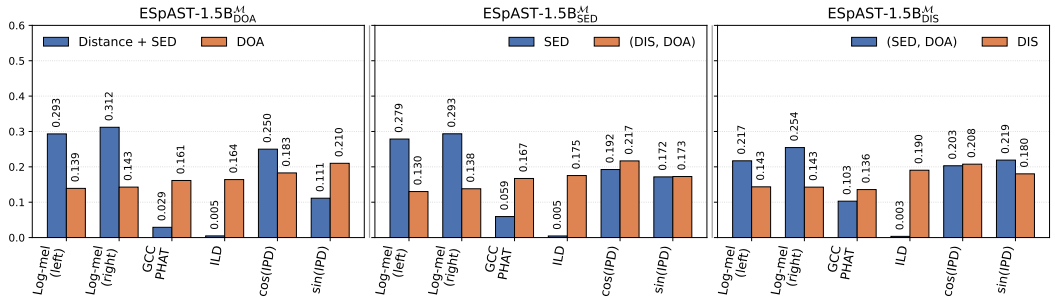

Figure 14: Normalized average input feature usage of the feature attention layer in ESpAST-1.5B$^{\mathcal{M}}_{DOA}$, ESpAST-1.5B$^{\mathcal{M}}_{SED}$ and ESpAST-1.5B$^{\mathcal{M}}_{DIS}$.

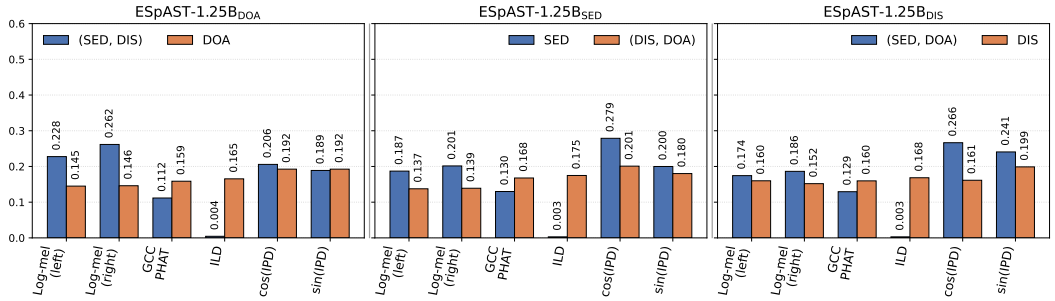

Figure 15: Normalized average input feature usage of the feature attention layer in ESpAST-1.25B$_{DOA}$, ESpAST-1.25B$_{SED}$ and ESpAST-1.25B$_{DIS}$.

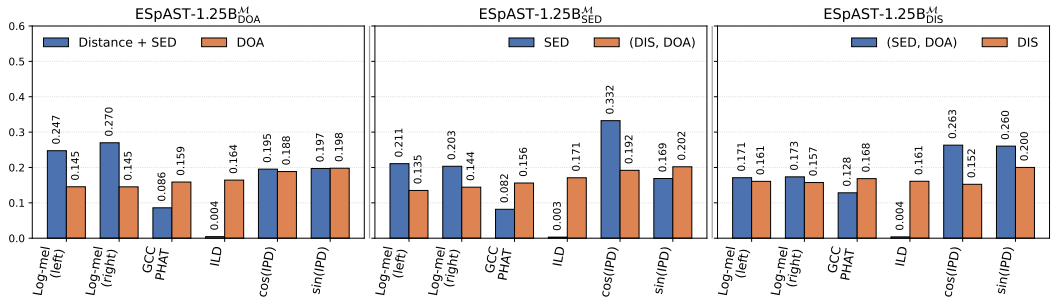

Figure 16: Normalized average input feature usage of the feature attention layer in ESpAST-1.25B$^{\mathcal{M}}_{DOA}$, ESpAST-1.25B$^{\mathcal{M}}_{SED}$ and ESpAST-1.25B$^{\mathcal{M}}_{DIS}$.