ES-Merging: Biological MLLM Merging via Embedding Space Signals

Wonbin Lee[*1] **Dongki Kim**[*1] **Sung Ju Hwang**[1,2]
KAIST[1], DeepAuto.ai[2]
{smilelwb01, cleverki, sungju.hwang}@kaist.ac.kr

Abstract

Biological multimodal large language models (MLLMs) have emerged as powerful foundation models for scientific discovery. However, existing models are specialized to a single modality, limiting their ability to solve inherently cross-modal scientific problems. While model merging is an efficient method to combine the different modalities into a unified MLLM, existing methods rely on input-agnostic parameter space heuristics that fail to faithfully capture modality specialization. To overcome this limitation, we propose the **E**mbedding-**S**ignal-based MLLM **Merging** (**ES-Merging**), a framework that estimates merging coefficients from embedding space signals, moving the merging paradigm from the parameter signals to the embedding signals. ES-Merging exploits coarse-grained and fine-grained signals from embedding space to estimate the layer-wise and element-wise merging coefficients, respectively, which are jointly combined for complementary coefficient estimation. Through extensive experiments, we demonstrate that ES-Merging outperforms existing merging methods not only on the cross-modal reasoning but also on the single-modal knowledge preservation, establishing that embedding space signals provide a principled and effective foundation for MLLM merging. Our code is available at https://github.com/WonbinLee067/es-merging.

1 Introduction

Multimodal Large Language Models (MLLMs) have been emerging as crucial foundational models for scientific discovery, extending their perception to diverse biological modalities across molecules [42, 29], proteins [1, 17], and cells [16, 46]. Despite their impressive progress within each modality, diverse scientific problems of interest are inherently cross-modal, requiring to understand the interactive effects such as protein-ligand interactions or drug effectiveness to cell types. However, existing biological MLLMs are specialized to a single modality, resulting in limited intersectional knowledge across diverse biological modalities and unreliable cross-modal reasoning.

Although model merging has been actively explored as an efficient scheme for integrating multiple models, how to merge modality-specialized MLLMs remains largely underexplored. In this setting, the central challenge is how to accurately measure modality specialization of an MLLM that induces the modality-specific behavior. Although modality specialization is ultimately encoded in model parameters, its functional effect is revealed through the representation transformations induced by modality-specific inputs. However, existing merging methods typically rely on input-blind parameter-space heuristics, such as magnitudes, signs, or directions of parameter updates. These signals can provide useful but indirect signals of specialization, since they do not observe how each expert responds to modality token embeddings. This motivates estimating merging coefficients from embedding-space signals that directly reflect input-conditioned modality behavior.

[*]Equal Contribution

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

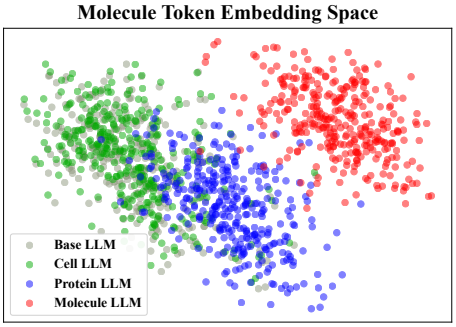


Figure 1: Molecule token embedding visualization of the last transformer block for the base LLM and each specialized LLM.

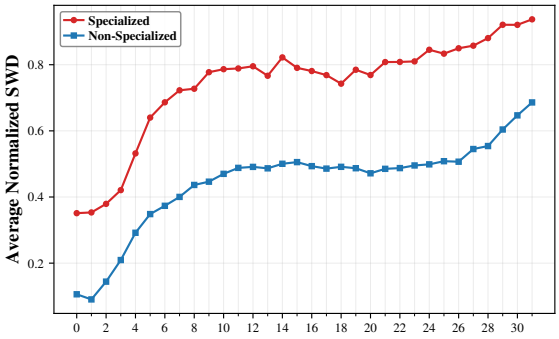


Figure 2: Layer-wise embedding distribution distance under specialized and non-specialized tokens using sliced Wasserstein distances (SWD) [2].

To examine whether embedding-space signals can indeed reveal modality specialization, we analyze how modality-specific inputs are processed by the base LLM and modality-specialized MLLMs. As shown in Fig. 1, when molecular tokens are processed by different MLLMs, the hidden representations form clearly different embedding distributions. In particular, the molecule LLM induces a more distinct distribution due to its modality-specific understanding. Further, we measure the embedding distribution distance between the base LLM and each specialized MLLM under specialized and non-specialized token inputs. Fig. 2 shows that specialized inputs consistently produce larger embedding distribution distance than non-specialized inputs, suggesting that the embedding space faithfully reflects modality specialization. Based on this analysis, we use embedding-space discrepancies induced by modality-token inputs as input-conditioned signals for estimating merging coefficients.

Motivated by this observation, we propose an embedding-signal-based MLLM merging (**ES-Merging**), a novel framework that moves the model merging paradigm from parameter space signals to embedding space signals. Our intuition is that the modality specialization of an MLLM can be measured by the differences in the embedding space between the base LLM and the MLLM as shown in Fig. 1. To this end, we first design a probe input containing multimodal tokens of different modalities and obtain layer-wise embeddings that reflect modality-specific representation changes across layers. Based on the embedding differences between each MLLM and base LLM, we estimate merging coefficients at two complementary granularity levels: layer-wise and element-wise. The layer-wise coefficients capture coarse-grained modality specialization across transformer layers, whereas the element-wise coefficients capture fine-grained specialization within each layer. By integrating these complementary signals, ES-Merging produces calibrated merging coefficients that determine how much each modality-specialized MLLM contributes to the unified MLLM.

We validate ES-Merging by merging three modality-specialized biological MLLMs, covering molecule, protein, and cell modalities, into a single unified model. Through extensive experiments, we demonstrate that ES-Merging substantially outperforms parameter-signal-based merging methods on cross-modal reasoning as well as on single-modal knowledge preservation. Further analyses show that ES-Merging is an efficient counterpart to task-specific fine-tuning by achieving a 5.95$\times$ reduction in computational cost, while providing explainability along with stronger or comparable performance. Additional experiments on image, video, and audio modalities suggest that embedding space signals can provide useful guidance beyond the biological domain. These results highlight that the embedding space provides principled and efficient signals for merging MLLMs.

2 Related Work

MLLMs for Science Large language models (LLMs) [49, 18, 40, 41, 10] are increasingly being extended to scientific problems through multimodal large language models (MLLMs) that incorporate diverse biological modalities, including molecules [54], proteins [59], and cells [11]. In the protein domain, prior works have modeled amino-acid sequences [60, 43] or jointly with structures [19, 1, 51, 58, 53, 17]. Single cell LLMs learn the scRNA-seq representations [15, 28, 16] or further incorporate histology information [33, 6]. On the other hand, molecular LLMs have been built on 1D string representations [9, 13] such as SMILES [55] and SELFIES [31], 2D molecular graphs [37, 14, 3,

64, 42], 3D structures with conformations [34, 20], and joint 2D-3D representations with multiple encoders [29]. Despite their progress, existing biological MLLMs are limited to understand and process a single modality, hindering their ability to solve inherently cross-modal scientific problems.

Model Merging Model merging aims to integrate multiple specialist models into a unified model with minimal additional training. Existing methods typically guide merging through parameter-space signals, including magnitude or agreement of parameter updates [27, 52, 12], sign alignment or masking for resolving update conflicts [61, 25], learned layer-wise coefficients [62], input-dependent routing [39], and latent-space interpolation [48]. Recent MLLM merging methods also remain largely grounded in parameter- or weight-space operations, such as improving the robustness of low-rank module merging [66] or using uncertainty to select and order vision-language MLLMs before weight-space merging [44]. However, these methods do not directly estimate modality specialization from embedding space responses induced by modality-specific token inputs. We instead propose a merging framework that derives layer-wise and element-wise merging coefficients from embedding signals, enabling modality-aware merging of biological MLLMs specialized to heterogeneous modalities.

3 Preliminary

We begin by formally describing MLLMs, then formulating the model merging with LoRA.

Multimodal Large Language Model MLLMs operate beyond the textual space by taking modality token embeddings with text tokens. Concretely, let $\mathcal{M} = \{m_1, ..., m_K\}$ denote a set of modalities. For each modality m_i, a modality-specific encoder f_{m_i} projects the raw modality input $\mathbf{x}_{m_i}$ to a sequence of modality tokens: $\mathbf{H}_{m_i} = f_{m_i}(\mathbf{x}_{m_i})$. Then, an MLLM generates textual output $\mathbf{y}$ by taking the concatenated textual and modality tokens: $\mathbf{y} = g(\mathbf{H}^0),$ where $\mathbf{H}^0 = [\mathbf{H}_{\text{text}}; \mathbf{H}_{m_i}; ...]$. In this view, the core interface of an MLLM is the token-based embeddings: raw modality inputs are first mapped into sequences of vectors that are compatible with the LLM's embedding dimension, and then transformed over layers to elicit modality-specific behaviors.

LoRA Merging In the MLLM setting, modality specialization is often implemented with LoRA [23], enabling unified model construction via weighted LoRA merging. Let θ_{m_i} denote the LoRA parameters specialized to modality m_i. We index the parameters by layer and weight element: $\theta^{l,n}_{m_i}$ denotes the n-th weight in the l-th layer of θ_{m_i}. We obtain the unified LoRA parameters as a weighted summation: $\theta^{l,n}_{\text{uni}} \leftarrow \sum_{m_i \in \mathcal{M}} \lambda^{l,n}_{m_i} \theta^{l,n}_{m_i}$ [24]. Under this formulation, the main challenge is to estimate appropriate merging coefficients $\lambda^{l,n}_{m_i}$ that determine how strongly each modality-specific parameter contributes to modality understanding. Our method addresses this coefficient estimation problem by deriving $\lambda^{l,n}_{m_i}$ from embedding space signals rather than parameter space statistics.

4 ES-Merging

We present Embedding-Signal-based MLLM Merging (ES-Merging), a representation-aware framework for merging modality-specialized MLLMs based on embedding space signals. Specifically, ES-Merging constructs probe inputs to elicit representational differences across models, and converts these embedding space discrepancies into layer-wise and element-wise merging coefficients.

4.1 Probe Input

Probe inputs are designed to provide a common diagnostic context in which the base LLM and all modality-specialized MLLMs process the same set of modality-token embeddings. Rather than using downstream task prompts, which may introduce task-specific biases, we construct probe inputs only to elicit the representation transformations induced by modality tokens. This allows us to measure modality specialization from the backbone response itself, while keeping the input context shared across different specialized models. Specifically, for each modality m_i, we sample raw inputs $\mathbf{x}_{m_i}$ from the corresponding dataset and obtain modality token embeddings as $\mathbf{H}_{m_i} = f_{m_i}(\mathbf{x}_{m_i})$. We construct each probe input by concatenating short textual prefixes with modality token blocks, i.e., $\mathbf{H}^0_{\text{probe}} = [\mathbf{H}_{\text{text},m_1}; \mathbf{H}_{m_1}; \mathbf{H}_{\text{text},m_2}; \mathbf{H}_{m_2}; \dots]$, where $\mathbf{H}_{\text{text},m_i}$ denotes the text-prefix embeddings.

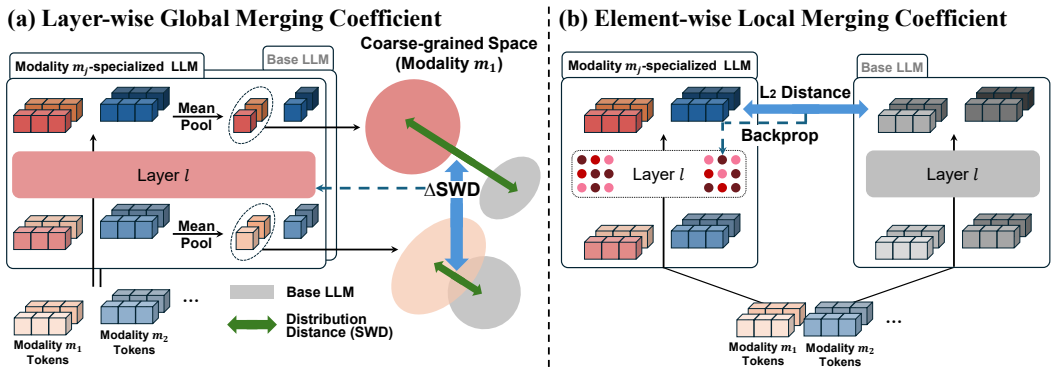


Figure 4: Overview of ES-Merging. (A) Layer-wise global merging coefficients are computed from the coarse-grained embedding signals, which are the layer-wise differences of distribution distances between mean pooled representations of the base LLM and a specialized MLLM. (B) Element-wise local merging coefficients are assigned based on the fine-grained embedding signals by computing the gradients from the embedding-wise distances.

In the biological setting, the modality set $\mathcal{M}$ typically consists of {molecule, protein, cell}, and the resulting probe input is illustrated in Fig. 3. By forwarding each probe input through the base LLM and each MLLM specialized to modality m_j, we obtain layer-wise embeddings by extracting the modality m_i tokens, denoted as $\mathbf{H}^l_{m_i \to \text{base}}$ and $\mathbf{H}^l_{m_i \to \theta_{m_j}}$.

```
Molecule: <molecule_tokens>
Protein: <protein_tokens>
Cell: <cell_tokens>
```

Figure 3: Probe input template.

These layer-wise representations serve as the basis for estimating merging coefficients. We regard the embedding discrepancy between the base LLM and each modality-specialized MLLM as an embedding space signal that reflects the degree of modality specialization. ES-Merging uses this signal at two complementary levels: globally across layers to estimate layer-wise specialization, and locally within each layer to identify parameter elements associated with specialized transformations.

4.2 Layer-wise Global Merging Coefficient

We first estimate the layer-wise contribution of each modality-specialized MLLM to the merged model from coarse-grained embedding signals (Fig. 4 (a)). We aim to assign a larger merging coefficient when a layer induces stronger representational transformations from the base LLM across modality inputs, reflecting the substantial modality-specific transformations.

Coarse-grained Embedding Signal Given the layer-wise modality-token embeddings from the probe inputs, $\mathbf{H}^l_{m_i \to \text{base}}$ and $\mathbf{H}^l_{m_i \to \theta_{m_j}}$, we first apply mean pooling over the token dimension to summarize each representation. This gives a coarse-grained representation of how the base LLM and each modality-specialized MLLM process the same modality tokens at each layer. We then measure the distributional gap between the representations of the base LLM and specialized MLLMs using sliced Wasserstein distance (SWD) [2]. Since the SWD at layer l reflects the cumulative representational shift up to that layer, we use the layer-wise differences, $\Delta\text{SWD}^l_{m_i \to \theta_{m_j}} = \text{SWD}^l_{m_i \to \theta_{m_j}} - \text{SWD}^{l-1}_{m_i \to \theta_{m_j}}$, as the coarse-grained embedding signal. A larger value indicates that the corresponding layer introduces a stronger modality-specific transformation.

Layer-wise Coefficient Estimation To convert the coarse-grained signals into merging coefficients, we normalize the layer-wise SWD differences and then aggregate them over input modalities to obtain the layer-wise importance score of each specialized model, defined as $s^l_{\theta_{m_j}} = \sum_{m_i \in \mathcal{M}} \widehat{\Delta\text{SWD}}^l_{m_i \to \theta_{m_j}}$. This score reflects how strongly θ_{m_j} contributes to coarse-grained representational transformations over different modality inputs at layer l. We then apply a softmax across specialized models to obtain the layer-wise global merging coefficient: $\alpha^l_{m_j} = \frac{\exp(s^l_{\theta_{m_j}}/\tau)}{\sum_{m \in \mathcal{M}} \exp(s^l_{\theta_m}/\tau)}$.

Thus, each specialized MLLM receives a larger layer-wise coefficient at layers where it induces stronger coarse-grained representational changes.

4.3 Element-wise Local Merging Coefficient

While the layer-wise global coefficients capture the contribution of each specialized model at the transformer-layer level, it assigns the merging coefficient uniformly to all parameters within a layer. To capture fine-grained contributions of each parameter, we further introduce an element-wise local merging coefficient from fine-grained embedding signals (Fig. 4 (b)).

Fine-grained Embedding Signal Unlike the coarse-grained signal, which is computed from mean-pooled representations, the fine-grained signal directly measures discrepancies between modality-token representations. For each modality m_i and specialized model θ_{m_j}, we compute the Frobenius distance between $\mathbf{H}^l_{m_i \to \text{base}}$ and $\mathbf{H}^l_{m_i \to \theta_{m_j}}$, averaged over the probe inputs. We denote this fine-grained discrepancy as $r^l_{m_i \to \theta_{m_j}}$. This signal reflects how differently the specialized model processes modality-m_i tokens relative to the base LLM at layer l.

Element-wise Coefficient Estimation To derive the parameter-level merging coefficients, we identify which parameter most affect to the fine-grained embedding discrepencies between base LLM and the specialized MLLMs, $r^l_{m_i \to \theta_{m_j}}$. Specifically, we regard $r^l_{m_i \to \theta_{m_j}}$ as a local alignment signal and, by computing the gradient of this discrepancy with respect to each parameter element, we obtain the element-wise importance score, $s^{l,n}_{\theta_{m_j}} = \sum_{m_i \in \mathcal{M}} \left| \partial r^l_{m_i \to \theta_{m_j}} / \partial \theta^{l,n}_{m_j} \right|$. A larger score indicates that the corresponding element has a stronger local influence on the base-specialized embedding gap, and thus is more closely tied to modality-specialized representation changes. We normalize theses scores and then apply a softmax across specialized models to obtain the element-wise local merging coefficient: $\beta^{l,n}_{m_j} = \frac{\exp(\hat{s}^{l,n}_{\theta_{m_j}} / \tau)}{\sum_{m \in \mathcal{M}} \exp(\hat{s}^{l,n}_{\theta_m} / \tau)}$. Each parameter element receives a larger coefficient when it more strongly governs fine-grained embedding-space discrepancies.

4.4 Integrating Layer- and Element-wise Merging Coefficients

The coarse-grained layer-wise coefficient $\alpha^l_{m_i}$ and the fine-grained element-wise coefficient $\beta^{l,n}_{m_i}$ each capture modality-specific importance at different levels of granularity. To complementarily combine two different coefficients, we compute a final merging coefficient by multiplying the layer-wise coefficient and the element-wise coefficient and renormalizing across modalities as follows:

$$\lambda^{l,n}_{m_i} = \frac{\alpha^l_{m_i} \cdot \beta^{l,n}_{m_i}}{\sum_{m \in \mathcal{M}} \alpha^l_m \cdot \beta^{l,n}_m}.$$

By integrating different coefficient types from different granularities of embedding signals, ES-Merging enables calibrated merging coefficients that preserve complementary modality expertise more faithfully, enabling robust cross-modal knowledge composition.

5 Experimental Results

5.1 Experimental Setup

Implementation Details For merging, we leverage the state-of-the-art MLLMs specialized to each modality: Mol-LLaMA [29] for molecule modality, Prot2Text-V2 [17] for protein modality, and Cell-o1 [16] for single cell modality, whose base LLMs are LLaMA-3.1-8B-Instruct [18]. The merging methods are applied across all modality-specific models, resulting in a unified MLLM. Since each downstream task requires distinct domain knowledge, we provide task-specific few-shot in-context examples to induce the appropriate task understanding. For a fair comparison, we use the same instruction templates and in-context examples for all compared methods and leverage the greedy decoding during the LLM inference to reduce randomness. For more details about the implementations and the prompts, please refer to Appendix A.3 and A.4, respectively.

Table 1: Performance comparison of ES-Merging with the base, specialized MLLMs, and merging methods on the instance-varying interaction prediction tasks. We report accuracy and macro-F1 across each subset. **Bold** and underline indicate the best and second best performances, respectively.

	Molecule-Protein Interaction								Molecule-Cell Interaction					
	BindingDB		BioSNAP		Human		*Avg.*		DrugComb		GDSC2		*Avg.*	
	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1
Base LLM and Specialized MLLMs														
LLaMA-3.1-8B-Instruct	51.9	51.7	61.5	60.2	59.0	58.8	57.5	56.9	73.7	73.6	84.8	84.7	79.3	79.1
Mol-LLaMA [29]	55.8	54.7	66.4	64.0	61.5	61.4	61.2	60.0	64.1	64.1	55.5	55.5	59.8	59.8
Prot2Text-V2 [17]	59.2	59.2	55.3	54.1	47.2	47.2	53.9	53.5	65.6	65.6	59.7	57.9	62.6	61.7
Cell-o1 [16]	53.5	51.5	59.3	59.3	49.1	48.0	54.0	52.9	76.3	76.2	85.9	85.8	81.1	81.0
Merging Methods														
Avg. Merging	65.3	64.9	67.4	66.5	60.9	60.9	64.5	64.1	72.5	72.4	85.4	85.2	78.9	78.8
TIES-Merging [61]	60.8	60.7	62.7	61.6	58.6	58.5	60.7	60.3	74.7	74.6	85.9	85.7	80.3	80.2
EMR-Merging [25]	64.7	64.2	67.3	66.8	60.4	60.4	64.1	63.8	45.2	45.0	93.4	93.3	69.3	69.2
AdaMerging [62]	52.2	51.0	64.2	61.7	59.9	59.8	58.8	57.5	16.0	14.1	17.2	15.3	16.6	14.7
PCB-Merging [12]	55.1	55.1	60.3	58.8	58.6	58.4	58.0	57.4	77.3	77.3	86.0	85.8	81.7	81.6
Consensus Merging [52]	59.2	59.2	62.1	60.9	59.2	59.1	60.2	59.7	76.0	76.0	84.3	84.2	80.2	80.1
UQ-Merge [44]	43.7	40.7	56.5	51.1	49.3	48.4	49.8	46.8	14.0	13.8	71.7	66.5	42.8	40.1
RobustMerge [66]	42.8	42.8	49.0	48.2	43.9	43.4	45.2	44.8	11.0	10.8	9.5	9.4	10.3	10.1
LS-Merge [48]	52.3	52.1	61.8	60.6	58.8	58.5	57.6	57.1	77.3	77.3	84.5	84.4	80.9	80.8
Avg. Merging + FT	60.5	59.5	55.8	55.8	57.2	57.2	57.8	57.5	**81.1**	**80.8**	94.0	93.9	**87.5**	**87.4**
ES-Merging (Ours)	**66.0**	**65.3**	**69.1**	**68.4**	**62.0**	**61.9**	**65.7**	**65.2**	80.7	80.2	**94.1**	**94.0**	87.4	87.1

Baselines We compare ES-Merging with the base LLM, MLLMs specialized to a single modality, and merging methods. As modality-specialized baselines, we consider Mol-LLaMA, Prot2Text-V2, and Cell-o1, each specialized for molecule, protein, and single-cell modalities, respectively. We also compare ES-Merging with parameter-signal-based merging methods, including Average Merging, TIES-Merging [61], EMR-Merging [25], layer-wise AdaMerging [62], PCB-Merging [12], Consensus Merging [52], LS-Merge [48], and task-specific finetuned model after the average merging, denoted as Avg. Merging + FT. For the base LLMs and the specialized MLLMs, unsupported modalities are represented as textual inputs. Baseline details are provided in Appendix A.2.

5.2 Instance-varying Interaction Prediction

We first evaluate on instance-varying cross-modal interaction tasks, where the target counterpart changes across instances and the model should generalize across diverse cross-modal pairs.

Datasets We consider two instance-varying cross-modal interaction settings: molecule-protein and molecule-cell interactions. For molecule-protein interaction, the task is to predict whether a given molecule interacts with a given protein, including BindingDB, BioSNAP, and Human [30]. For molecule-cell interaction, the task is to predict the effect of molecules on a given cell, including DrugComb [65] and GDSC2 [4]. In both settings, the molecule, protein, and cell counterpart changes for each instance, requiring the model to generalize across diverse cross-modal combinations rather than relying on a fixed target. We provide the detailed explanation of datasets in Appendix A.1.

Results As shown in Table 1, ES-Merging consistently outperforms parameter-signal-based merging baselines, suggesting that ES-Merging effectively integrates complementary cross-modal knowledge from modality-specialized MLLMs and thus leading to stronger generalization when the interaction counterpart varies across instances. Interestingly, ES-Merging outperforms the task-specific finetuned model (Avg. Merging + FT) on the molecule-protein interaction tasks and shows comparable performance on the molecule-cell interaction tasks, indicating that ES-merging can enhance cross-modal reasoning without further downstream finetuning. On the other hand, existing merging baselines such as EMR-Merging or RobustMerge often exhibit instability across datasets. These results imply that input-blind parameter space heuristics are insufficient for reliably combining heterogeneous multimodal specialization, supporting that ES-Merging provides a more robust integration of modality-specific knowledge and behaviors from the input-aware embedding space signals.

Table 2: Performance comparison of ES-Merging with the base, specialized MLLMs, and merging methods on the target-fixed functionality prediction tasks. We report accuracy and macro-F1 across each subset. **Bold** indicates the best and underline indicates the second best.

	CYP Inhibition												CYP Substrate							
	CYP1A2		CYP2C19		CYP2C9		CYP2D6		CYP3A4		*Avg.*		CYP2C9		CYP2D6		CYP3A4		*Avg.*	
	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1
Base LLM and Specialized MLLMs																				
LLaMA-3.1-8B-Instruct	58.9	55.3	56.0	50.6	55.9	47.4	61.2	56.4	55.1	43.1	57.4	50.6	57.7	47.8	34.2	30.1	55.6	48.5	49.2	42.1
Mol-LLaMA	68.6	67.1	65.8	64.3	63.3	63.3	64.7	59.2	64.3	64.0	65.3	63.6	62.7	49.7	**61.7**	**57.4**	55.2	53.7	59.9	53.6
Prot2Text-V2	59.4	55.8	56.8	52.5	67.4	60.1	76.3	55.9	59.6	50.3	63.9	54.9	54.5	47.5	53.4	51.4	56.0	53.8	54.6	50.9
Cell-o1	53.5	53.5	44.9	44.9	43.2	40.6	46.9	40.9	43.6	43.5	46.4	44.7	36.6	34.3	36.1	34.9	48.5	47.5	40.4	38.9
Merging Methods																				
Avg. Merging	61.2	60.4	54.1	53.2	52.7	52.5	53.8	49.5	54.6	54.6	55.3	54.0	40.3	38.6	42.1	41.3	49.3	45.8	43.9	41.9
TIES-Merging	71.8	71.7	67.8	67.7	66.5	63.5	76.1	63.5	65.3	64.8	69.5	66.2	56.0	43.5	55.6	52.1	54.5	53.1	55.4	49.6
EMR-Merging	70.4	70.4	64.8	64.7	66.8	64.5	74.2	61.7	66.1	65.6	68.5	65.4	60.5	49.2	55.6	52.9	51.5	49.4	55.9	50.5
AdaMerging	52.5	49.4	53.1	50.3	44.8	43.6	45.6	43.9	50.5	49.4	49.3	47.3	33.6	33.6	43.6	43.4	49.3	45.3	42.2	40.8
PCB-Merging	69.2	69.0	63.6	63.3	65.0	61.8	72.4	60.1	63.8	63.0	66.8	63.4	55.2	44.9	57.1	53.3	56.7	55.0	56.4	51.1
Consensus Merging	68.5	68.2	63.6	63.3	65.7	62.3	73.5	60.2	63.8	62.9	67.0	63.4	54.5	45.3	55.6	52.9	56.7	55.7	55.6	51.3
UQ-Merge	50.3	49.5	48.1	36.4	36.2	29.8	26.1	24.9	42.7	32.6	40.5	34.6	26.1	24.9	28.6	26.9	49.3	49.3	34.7	33.7
RobustMerge	41.2	40.6	39.6	39.4	36.6	36.6	39.2	35.9	40.8	40.8	39.5	38.7	50.0	46.8	45.9	45.3	44.8	41.9	46.9	44.7
LS-Merge	55.3	55.3	51.0	51.0	47.6	46.3	56.1	48.5	44.1	44.1	50.8	49.0	44.0	40.9	39.1	38.4	41.8	39.8	41.6	39.8
Avg. Merging + FT	68.3	68.3	67.0	67.0	62.3	62.2	67.6	67.6	67.7	67.6	66.6	66.5	**65.7**	50.6	59.4	53.0	55.2	54.6	60.1	52.7
ES-Merging (Ours)	**77.4**	**77.4**	**70.6**	**70.5**	**72.5**	**70.4**	**80.7**	**69.5**	**71.3**	**70.8**	**74.5**	**71.7**	64.2	**53.6**	60.9	57.2	**60.5**	**59.6**	**61.9**	**56.8**

5.3 Target-fixed Functionality Prediction

We further evaluate on target-fixed cross-modal functionality prediction tasks, where each subtask is associated with a fixed target and requires target-specific biological functionality knowledge beyond simple interaction matching in Section 5.2.

Datasets To this end, we evaluate on CYP enzyme prediction. Unlike the instance-varying setting above, each subtask is defined with respect to a fixed enzyme target and the tasks are predicting the biological functionality of the given molecule to the fixed target: inhibitory effects or substrate specificity. Specifically, we consider five CYP inhibition subtasks to CYP1A2, CYP2C19, CYP2C9, CYP2D6, and CYP3A4 enzymes [56], and three CYP substrate subtasks for CYP2C9, CYP2D6, and CYP3A4 enzymes [22], from the TDC dataset [26]. These benchmarks therefore evaluate whether the merged model can capture not only detailed molecular structural variation under a shared target, but also biologically meaningful interaction types.

Results As shown in Table 2, ES-Merging achieves the best average performance, suggesting that it more effectively preserves and integrates expert knowledge required for target-specific functionality prediction. In contrast, on the CYP substrate tasks, most merging baselines based on the parameter signals underperform Mol-LLaMA, because these tasks largely rely on molecular structural understanding under a fixed-target setting. ES-Merging achieves performance comparable to or better than Mol-LLaMA, indicating that embedding-signal-based merging preserves molecular structural understanding while incorporating target-specific functional knowledge from expert models.

5.4 Single-modal Knowledge Preservation

To see whether ES-Merging can preserve the knowledge that is learned by each specialized MLLMs, we evaluate merging methods on single-modal tasks beyond cross-modal tasks. We leverage PAMPA [50], Immuno-Virus [35], and PBMC [38] as single-modal tasks related to molecules, proteins, and cells, respectively. For detailed experimental setup, please refer to Appendix A.4.

Results As shown in Table 3, ES-Merging substantially improves single-modal knowledge preservation over existing merging methods, achieving the highest average accuracy among all merged models, indicating that embedding space signals help mitigate the degradation of modality-specific knowledge that commonly occurs when heterogeneous experts are merged in parameter space. Notably, ES-Merging marginally surpasses the corresponding specialized MLLMs on the molecule and protein tasks. These results suggest that ES-Merging not only preserves expert knowledge, but also constructs a more effective shared representation by integrating complementary information from other modality-specialized models. On the cell task, ES-Merging achieves the best performance

Table 3: Performance comparison on single-modality tasks. We report the accuracy for each task with the average accuracy.

	Molecule	Protein	Cell	*Avg.*
Specialized MLLM	75.68	85.25	77.84	79.59
Avg. Merging	64.34	70.87	55.94	63.72
TIES-Merging [61]	45.67	62.30	46.90	51.62
EMR-Merging [25]	61.22	61.03	50.13	57.46
AdaMerging [62]	61.33	73.52	53.06	62.64
PCB-Merging [12]	49.99	57.88	50.14	52.67
Consensus Merging [52]	48.39	67.59	46.94	54.31
UQ-Merge [44]	39.56	79.32	52.42	57.10
RobustMerge [66]	49.14	64.19	55.16	56.16
LS-Merge [48]	48.31	59.14	49.13	52.19
ES-Merging (Ours)	**77.64**	**86.51**	**61.49**	**75.21**

Table 4: Total FLOPs until determining merged LoRA parameters.

Method	**Total FLOPs** ↓
Avg. Merging + FT	891,117
AdaMerging	493,694
ES-Merging (Ours)	**149,807**

Table 5: Ablation studies on different merging coefficient types.

Task	**L**	**E**	**L×E**
Mol-Prot	63.6	64.9	**65.7**
Mol-Cell	85.2	86.7	**87.4**
CYP Inhibit	73.9	72.7	**74.5**
CYP Subs.	57.1	60.5	**61.9**

among all merging methods, although it remains below the cell-specialized MLLM. This is mainly due to the little transferability of non-cell MLLMs to cell-related tasks as discussed in Appendix C.2, explaining why cell performance is more susceptible to merging-induced interference. Nevertheless, ES-Merging alleviates this degradation more effectively than existing merging approaches.

5.5 Analysis

ES-Merging as an Efficient and Explainable Alternative to Fine-Tuning We further compare ES-Merging with task-specific fine-tuning from the perspectives of qualitative response behavior and computational efficiency. As shown in Tables 8 and 9 of Appendix, both ES-Merging and the task-specific fine-tuned model predict the correct labels in the analyzed cases. Interestingly, ES-Merging provides biological rationales that explain its predictions by connecting features and information across different modalities to the target task, whereas the task-specific fine-tuned model outputs only the final label, showing that ES-Merging can retain more explicit cross-modal reasoning behavior learned by each specialized MLLM, rather than producing task labels.

ES-Merging is also substantially more efficient than optimization-based alternatives. As shown in Table 4, ES-Merging reduces the computational cost of determining the merged LoRA parameters by 5.95× compared to task-specific fine-tuning and by 3.30× compared to AdaMerging. This efficiency comes from the fact that ES-Merging estimates merging coefficients once from probe-induced embedding signals, whereas task-specific fine-tuning and AdaMerging require iterative gradient-based optimization. These observations suggest that ES-Merging provides a practical alternative to task-specific fine-tuning, offering explicit biological rationales and avoiding expensive training while achieving the best or comparable performance over diverse cross-modal reasoning tasks.

Ablation Study We further conduct an ablation study to see the effectiveness of each coefficient type. As shown in Table 5, using only one of the coefficient types already outperforms most merging baselines, indicating that embedding space signals provide salient and robust information for cross-modal merging compared to parameter-signal-based merging methods. Combining these two coefficients consistently yields the best performance across all task groups, suggesting that effective MLLM merging requires both coarse-grained layer selection and fine-grained parameter selection: the former prevents globally important layers from being diluted, while the latter avoids assigning uniform weights to all parameters within a layer.

Merging Coefficient Analysis We further visualize the layer-wise merging coefficients in Fig. 5 and the element-wise merging coefficients in Fig. 10 of Appendix. The coefficients are distinctly distributed across different MLLMs, showing that modality-specific knowledge is not incorporated uniformly throughout the model. Additionally, we observe that even when the layer-wise coefficient is high (e.g., the third layer of the molecule LLM), the element-wise merging coefficients within that layer vary substantially, as shown in Fig. 6. This within-layer heterogeneity indicates that a high layer-wise coefficient alone does not imply that all parameters in the layer should be merged with the

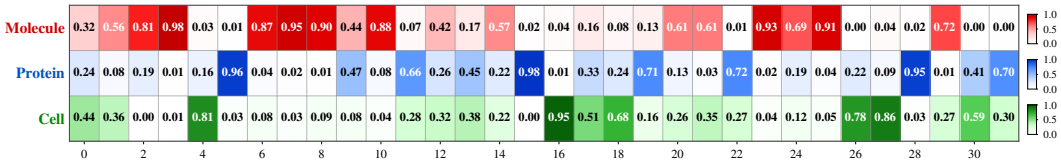


Figure 5: Computed layer-wise merging coefficient visualization of each specialized MLLM. The l-th column corresponds to the merging coefficient of the l-th layer, $\alpha_{m_j}^{l}$.

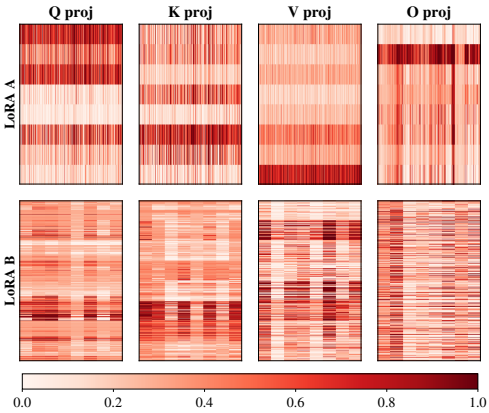


Figure 6: Element-wise merging coefficient visualization of the third layer in molecule LLM.

Table 6: Performance comparison on the MUSIC-AVQA task across different modality combinations.

Method	V	I	A	V + I	V + A	I + A	V + I + A
Base LLM and Specialized MLLMs							
Vicuna-7B	25.74	30.52	18.89	22.07	26.96	24.65	30.98
VideoLLM	49.13	47.07	40.35	49.41	48.91	48.56	51.49
ImageLLM	49.67	50.78	37.74	50.16	49.74	51.80	51.27
AudioLLM	30.84	43.07	28.28	29.55	29.61	37.69	30.75
Merging Methods							
Avg. Merging	46.82	48.22	40.64	45.35	47.25	48.65	47.30
TIES-Merging [61]	32.31	36.00	22.59	27.69	34.16	30.32	35.96
EMR-Merging [25]	52.18	50.77	41.11	52.28	51.82	52.16	53.58
AdaMerging [62]	49.73	50.54	43.18	49.63	50.21	52.30	52.18
PCB-Merging [12]	29.26	20.73	20.73	25.14	31.32	27.49	34.10
Consensus Merging [52]	32.41	21.96	21.96	27.91	33.73	29.18	35.77
UQ-Merge [44]	50.77	48.68	37.74	50.41	50.65	49.52	51.62
RobustMerge [66]	51.64	50.05	42.09	51.80	51.61	51.03	52.93
ES-Merging (Ours)	**53.69**	**53.11**	**44.10**	**53.76**	**53.06**	**53.89**	**53.59**

same strength. Instead, ES-Merging assigns larger weights only to the parameter elements that are more sensitive to modality-specific representation shifts, preventing irrelevant or weakly specialized parameters from being over-emphasized. Therefore, modality specialization emerges at multiple levels of granularity: only a few parameter elements within important layers are primarily salient for modality specialization. Therefore, combining layer-wise global coefficients with element-wise local coefficients leads to the complementary merging of different granularities, achieving an accurate and robust merging of different modalities.

General Domain Application Although ES-Merging is primarily designed and evaluated for biological MLLM merging, its core principle is applicable to diverse domains: merging coefficients are estimated from embedding-space discrepancies rather than modality-specific assumptions. To examine whether this principle can extend beyond biological modalities, we additionally evaluate ES-Merging on the MUSIC-AVQA [32] benchmark which requires understanding visual and acoustic signals from video, image, and audio. As shown in Table 6, ES-Merging achieves the best performance across all modality combinations, including single-, two-, and three-modality inputs. In particular, ES-Merging consistently outperforms parameter space merging baselines, suggesting that embedding space signals can also provide useful guidance for merging specialists in general multimodal domains. We also report additional experiments on AVQA in Appendix C.8.

Merging Coefficient Contribution Validation To directly examine whether embedding-space discrepancy reflects actual task contribution, we selectively remove layers and elements assigned high molecule-specific coefficients. For the layer-wise-only variant, we remove molecule-expert parameters from the eight layers whose molecule coefficients exceed 0.8. For the element-wise-only variant, we remove the 6.255% of all evaluated elements whose molecule coefficients exceed the same threshold. As shown in Table 7, both interventions substantially degrade performance, supporting that high embedding-based coefficients identify task-relevant expert contributions. We further examine whether the relative ordering induced by the element-wise scores is meaningful. We take the reciprocal of the coefficients to downweight high-score elements and emphasize low-score elements, followed by normalization across experts. Reversing the coefficient priorities reduces accuracy by 13.4 percentage points. This provides additional evidence that high element-wise scores capture task-relevant and beneficial expert contributions rather than harmful perturbations.

Table 7: Accuracy (%) on CYP2D6 inhibition under coefficient interventions.

Intervention	Original	Modified	Degradation (pp)
Remove high-coefficient layers	79.2	69.7	-9.5
Remove high-coefficient elements	77.6	62.3	-15.3
Reverse element-wise priority	77.6	64.2	-13.4

6 Conclusion

We present ES-Merging, a novel MLLM merging framework that shifts model merging from relying on parameter space signals to leveraging embedding space signals. Our core insight is that input-aware representations encode rich modality-specific specialization, providing a more faithful basis for integrating modality-specialized MLLMs. Based on this observation, we introduce two complementary types of merging coefficients at different granularities: layer-wise global coefficients for capturing coarse-grained specialization and element-wise local coefficients for capturing fine-grained importance. Experiments on diverse biological tasks demonstrate that ES-Merging consistently improves cross-modal reasoning while better preserving single-modal expertise. Further analyses show that ES-Merging is an efficient and explainable alternative to task-specific fine-tuning and that embedding space signals can also generalize to non-biological multimodal merging scenarios. We believe that our work establishes embedding space signals as a principled and practical foundation for constructing unified MLLMs from heterogeneous modality-specialized experts.

Acknowledgments

This work was supported by the Institute for Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (RS-2019-II190075, Artificial Intelligence Graduate School Program (KAIST)); the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2023-00256259); the "Advanced GPU Utilization Support Program" funded by the Government of the Republic of Korea (Ministry of Science and ICT); the Korea Machine Learning Ledger Orchestration for Drug Discovery Project (K-MELLODDY), funded by the Ministry of Health & Welfare and the Ministry of Science and ICT, Republic of Korea (grant number: RS-2024-00460870); the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2022-II220713, Meta-learning Applicable to Real-world Problems); and the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Ministry of Science and ICT (MSIT) of the Republic of Korea in connection with the Global AI Frontier Lab International Collaborative Research (No. RS-2024-00469482 & RS-2024-00509279).

References

[1] Hadi Abdine, Michail Chatzianastasis, Costas Bouyioukos, and Michalis Vazirgiannis. Prot2text: Multimodal protein's function generation with gnns and transformers. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 10757–10765, 2024.

[2] Nicolas Bonneel, Julien Rabin, Gabriel Peyré, and Hanspeter Pfister. Sliced and radon wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision*, 51(1):22–45, 2015.

[3] He Cao, Zijing Liu, Xingyu Lu, Yuan Yao, and Yu Li. Instructmol: Multi-modal integration for building a versatile and reliable molecular assistant in drug discovery. *arXiv:2311.16208*, 2023.

[4] Saurabh Chawla, Anja Rockstroh, Melanie Lehman, et al. Gene expression based inference of cancer drug sensitivity. *Nature Communications*, 13(1):5680, 2022. doi: 10.1038/s41467-022-33291-z.

[5] Chi Chen, Yiyang Du, Zheng Fang, Ziyue Wang, Fuwen Luo, Peng Li, Ming Yan, Ji Zhang, Fei Huang, Maosong Sun, et al. Model composition for multimodal large language models.

In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11246–11262, 2024.

[6] Chi-Jane Chen, Yuhang Chen, Sukwon Yun, Natalie Stanley, and Tianlong Chen. Spatial coordinates as a cell language: A multi-sentence framework for imaging mass cytometry analysis. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 13241–13252, 2025.

[7] Sanyuan Chen, Yu Wu, Chengyi Wang, Shujie Liu, Daniel Tompkins, Zhuo Chen, and Furu Wei. Beats: Audio pre-training with acoustic tokenizers. *arXiv preprint arXiv:2212.09058*, 2022.

[8] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. https://lmsys.org/blog/2023-03-30-vicuna/, March 2023.

[9] Seyone Chithrananda, Gabriel Grand, and Bharath Ramsundar. Chemberta: large-scale self-supervised pretraining for molecular property prediction. *arXiv:2010.09885*, 2020.

[10] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

[11] Sajib Acharjee Dip, Adrika Zafor, Bikash Kumar Paul, Uddip Acharjee Shuvo, Muhit Islam Emon, Xuan Wang, and Liqing Zhang. Llm4cell: A survey of large language and agentic models for single-cell biology. *arXiv preprint arXiv:2510.07793*, 2025.

[12] Guodong Du, Junlin Lee, Jing Li, Runhua Jiang, Yifei Guo, Shuyang Yu, Hanting Liu, Sim K Goh, Ho-Kin Tang, Daojing He, et al. Parameter competition balancing for model merging. *Advances in Neural Information Processing Systems*, 37:84746–84776, 2024.

[13] Carl Edwards, Tuan Lai, Kevin Ros, Garrett Honke, Kyunghyun Cho, and Heng Ji. Translation between molecules and natural language. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 375–413, 2022.

[14] Yin Fang, Xiaozhuan Liang, Ningyu Zhang, Kangwei Liu, Rui Huang, Zhuo Chen, Xiaohui Fan, and Huajun Chen. Mol-instructions: A large-scale biomolecular instruction dataset for large language models. *arXiv preprint arXiv:2306.08018*, 2023.

[15] Yin Fang, Xinle Deng, Kangwei Liu, Ningyu Zhang, Jingyang Qian, Penghui Yang, Xiaohui Fan, and Huajun Chen. A multi-modal ai copilot for single-cell analysis with instruction following. *arXiv preprint arXiv:2501.08187*, 2025.

[16] Yin Fang, Qiao Jin, Guangzhi Xiong, Bowen Jin, Xianrui Zhong, Siru Ouyang, Aidong Zhang, Jiawei Han, and Zhiyong Lu. Cell-o1: Training llms to solve single-cell reasoning puzzles with reinforcement learning. *arXiv preprint arXiv:2506.02911*, 2025.

[17] Xiao Fei, Michail Chatzianastasis, Sarah Almeida Carneiro, Hadi Abdine, Lawrence P Petalidis, and Michalis Vazirgiannis. Prot2text-v2: Protein function prediction with multimodal contrastive alignment. *Advances in Neural Information Processing Systems*, 2025.

[18] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[19] Han Guo, Mingjia Huo, Ruiyi Zhang, and Pengtao Xie. Proteinchat: Towards achieving chatgpt-like functionalities on protein 3d structures. *Authorea Preprints*, 2023.

[20] Shuhan Guo, Yatao Bian, Ruibing Wang, Nan Yin, Zhen Wang, and Quanming Yao. Unimot: Unified molecule-text language model with discrete token representation. *arXiv preprint arXiv:2408.00863*, 2024.

[21] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

[22] Malte Holmer, Christina de Bruyn Kops, Conrad Stork, and Johannes Kirchmair. Cypstrate: A set of machine learning models for the accurate classification of cytochrome p450 enzyme substrates and non-substrates. *Molecules*, 26(15):4678, 2021. doi: 10.3390/molecules26154678.

[23] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

[24] Chengsong Huang, Qian Liu, Bill Yuchen Lin, Tianyu Pang, Chao Du, and Min Lin. Lorahub: Efficient cross-task generalization via dynamic lora composition. *arXiv preprint arXiv:2307.13269*, 2023.

[25] Chenyu Huang, Peng Ye, Tao Chen, Tong He, Xiangyu Yue, and Wanli Ouyang. Emr-merging: Tuning-free high-performance model merging. *Advances in Neural Information Processing Systems*, 37:122741–122769, 2024.

[26] Kexin Huang, Tianfan Fu, Wenhao Gao, Yue Zhao, Yusuf Roohani, Jure Leskovec, Connor W Coley, Cao Xiao, Jimeng Sun, and Marinka Zitnik. Therapeutics data commons: Machine learning datasets and tasks for drug discovery and development. *arXiv preprint arXiv:2102.09548*, 2021.

[27] Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*, 2022.

[28] Oussama Kharouiche, Aris Markogiannakis, Xiao Fei, Michail Chatzianastasis, and Michalis Vazirgiannis. Cell2text: Multimodal llm for generating single-cell descriptions from rna-seq data. *arXiv preprint arXiv:2509.24840*, 2025.

[29] Dongki Kim, Wonbin Lee, and Sung Ju Hwang. Mol-llama: Towards general understanding of molecules in large molecular language model. *Advances in Neural Information Processing Systems*, 2025.

[30] Huan Yee Koh, Anh T.N. Nguyen, Shirui Pan, Lauren T. May, and Geoffrey I. Webb. Psichic: physicochemical graph neural network for learning protein-ligand interaction fingerprints from sequence data. *bioRxiv*, 2023. doi: 10.1101/2023.09.17.558145. URL https://www.biorxiv.org/content/early/2023/09/19/2023.09.17.558145.

[31] Mario Krenn, Florian Häse, AkshatKumar Nigam, Pascal Friederich, and Alan Aspuru-Guzik. Self-referencing embedded strings (selfies): A 100% robust molecular string representation. *Machine Learning: Science and Technology*, 1(4):045024, 2020.

[32] Guangyao li, Yake Wei, Yapeng Tian, Chenliang Xu, Ji-Rong Wen, and Di Hu. Learning to answer questions in dynamic audio-visual scenarios. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[33] Longyi Li, Liyan Dong, Hao Zhang, Dong Xu, and Yongli Li. spallm: enhancing spatial domain analysis in multi-omics data through large language model integration. *Briefings in Bioinformatics*, 26(4):bbaf304, 2025.

[34] Sihang Li, Zhiyuan Liu, Yanchen Luo, Xiang Wang, Xiangnan He, Kenji Kawaguchi, Tat-Seng Chua, and Qi Tian. Towards 3d molecule-text interpretation in language models. *arXiv preprint arXiv:2401.13923*, 2024.

[35] Song Li, Yang Tan, Song Ke, Liang Hong, and Bingxin Zhou. Immunogenicity prediction with dual attention enables vaccine target selection. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=hWmwL9gizZ.

[36] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Sal Candido, et al. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *BioRxiv*, 2022: 500902, 2022.

[37] Zhiyuan Liu, Sihang Li, Yanchen Luo, Hao Fei, Yixin Cao, Kenji Kawaguchi, Xiang Wang, and Tat-Seng Chua. MolCA: Molecular graph-language modeling with cross-modal projector and uni-modal adapter. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

[38] Romain Lopez, Jeffrey Regier, Michael B Cole, Michael I Jordan, and Nir Yosef. Deep generative modeling for single-cell transcriptomics. *Nature methods*, 15(12):1053–1058, 2018.

[39] Zhenyi Lu, Chenghao Fan, Wei Wei, Xiaoye Qu, Dangyang Chen, and Yu Cheng. Twin-merging: Dynamic integration of modular expertise in model merging. *Advances in Neural Information Processing Systems*, 37:78905–78935, 2024.

[40] OpenAI. Gpt-4 technical report. *arXiv:2303.08774*, 2024.

[41] OpenAI. Gpt-4o system card. *arXiv:2410.21276*, 2024.

[42] Jinyoung Park, Minseong Bae, Dohwan Ko, and Hyunwoo J Kim. Llamo: Large language model-based molecular graph assistant. *Advances in Neural Information Processing Systems*, 37:131972–132000, 2024.

[43] Qizhi Pei, Wei Zhang, Jinhua Zhu, Kehan Wu, Kaiyuan Gao, Lijun Wu, Yingce Xia, and Rui Yan. Biot5: Enriching cross-modal integration in biology with chemical knowledge and natural language associations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1102–1123, 2023.

[44] Huaizhi Qu, Xinyu Zhao, Jie Peng, Kwonjoon Lee, Behzad Dariush, and Tianlong Chen. Uq-merge: Uncertainty guided multimodal large language model merging. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 1401–1417, 2025.

[45] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[46] Syed Asad Rizvi, Daniel Levine, Aakash Patel, Shiyang Zhang, Eric Wang, Curtis Jamison Perry, Nicole Mayerli Constante, Sizhuang He, David Zhang, Cerise Tang, et al. Scaling large language models for next-generation single-cell analysis. *BioRxiv*, pages 2025–04, 2025.

[47] David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of chemical information and modeling*, 50(5):742–754, 2010.

[48] Bedionita Soro, Aoxuan Silvia Zhang, Bruno Andreis, Jaehyeong Jo, Song Chong, and Sung Ju Hwang. Ls-merge: Merging language models in latent space. In *The Fourteenth International Conference on Learning Representations*, 2026.

[49] Hugo Touvron et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv:2307.09288*, 2023.

[50] Alejandro Velez-Arce, Xiang Lin, Michelle Li, Kexin Huang, Wenhao Gao, Tianfan Fu, Bradley L Pentelute, Manolis Kellis, and Marinka Zitnik. Signals in the cells: Multimodal and contextualized machine learning foundations for therapeutics. *Advances in Neural Information Processing Systems Workshop on AI for New Drug Modalities*, 2024.

[51] Chao Wang, Hehe Fan, Ruijie Quan, and Yi Yang. Protchatgpt: Towards understanding proteins with large language models. *arXiv preprint arXiv:2402.09649*, 2024.

[52] Ke Wang, Nikolaos Dimitriadis, Guillermo Ortiz-Jimenez, François Fleuret, and Pascal Frossard. Localizing task information for improved model merging and compression. *arXiv preprint arXiv:2405.07813*, 2024.

[53] Zhicong Wang, Zicheng Ma, Ziqiang Cao, Changlong Zhou, Jun Zhang, and Yi Qin Gao. Prot2chat: protein large language model with early fusion of text, sequence, and structure. *Bioinformatics*, 41(8):btaf396, 2025.

[54] Ziqing Wang, Kexin Zhang, Zihan Zhao, Yibo Wen, Abhishek Pandey, Han Liu, and Kaize Ding. A survey of large language models for text-guided molecular discovery: from molecule generation to optimization. *arXiv preprint arXiv:2505.16094*, 2025.

[55] David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28 (1):31–36, 1988.

[56] Benjamin Weiser, Jérôme Genzling, Mihai Burai-Patrascu, Ophélie Rostaing, and Nicolas Moitessier. Machine learning-augmented docking. 1. cyp inhibition prediction. *Digital Discovery*, 2:1841–1849, 2023. doi: 10.1039/D3DD00110E. URL http://dx.doi.org/10.1039/D3DD00110E.

[57] Peter Willett, John M Barnard, and Geoffrey M Downs. Chemical similarity searching. *Journal of chemical information and computer sciences*, 38(6):983–996, 1998.

[58] Yijia Xiao, Edward Sun, Yiqiao Jin, Qifan Wang, and Wei Wang. Proteingpt: Multimodal llm for protein property prediction and structure understanding. *arXiv preprint arXiv:2408.11363*, 2024.

[59] Yijia Xiao, Wanjia Zhao, Junkai Zhang, Yiqiao Jin, Han Zhang, Zhicheng Ren, Renliang Sun, Haixin Wang, Guancheng Wan, Pan Lu, et al. Protein large language models: A comprehensive survey. *arXiv preprint arXiv:2502.17504*, 2025.

[60] Minghao Xu, Xinyu Yuan, Santiago Miret, and Jian Tang. Protst: Multi-modality learning of protein sequences and biomedical texts. In *International conference on machine learning*, pages 38749–38767. PMLR, 2023.

[61] Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. Ties-merging: Resolving interference when merging models. *Advances in neural information processing systems*, 36:7093–7115, 2023.

[62] Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. Adamerging: Adaptive model merging for multi-task learning. *arXiv preprint arXiv:2310.02575*, 2023.

[63] Pinci Yang, Xin Wang, Xuguang Duan, Hong Chen, Runze Hou, Cong Jin, and Wenwu Zhu. Avqa: A dataset for audio-visual question answering on videos. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 3480–3491, 2022.

[64] Botao Yu, Frazier N. Baker, Ziqi Chen, Xia Ning, and Huan Sun. Llasmol: Advancing large language models for chemistry with a large-scale, comprehensive, high-quality instruction tuning dataset. *arXiv preprint arXiv:2402.09391*, 2024.

[65] Bulat Zagidullin, Jehad Aldahdooh, Shuyu Zheng, Wenyu Wang, Yinyin Wang, Joseph Saad, Alina Malyutina, Mohieddin Jafari, Ziaurrehman Tanoli, Alberto Pessia, and Jing Tang. Drugcomb: an integrative cancer drug combination data portal. *Nucleic Acids Research*, 47(W1): W43–W51, 2019. doi: 10.1093/nar/gkz337.

[66] Fanhu Zeng, Haiyang Guo, Fei Zhu, Li Shen, and Hao Tang. Robustmerge: Parameter-efficient model merging for mllms with direction robustness. *arXiv preprint arXiv:2502.17159*, 2025.

[67] Bin Zhu, Bin Lin, Munan Ning, Yang Yan, Jiaxi Cui, HongFa Wang, Yatian Pang, Wenhao Jiang, Junwu Zhang, Zongwei Li, et al. Languagebind: Extending video-language pretraining to n-modality by language-based semantic alignment. *arXiv preprint arXiv:2310.01852*, 2023.

Appendix

Organization Appendix is organized as follows: In Section A, we provide detailed experimental settings, including datasets, baseline, implementation, and prompt setting details. In Section B, we qualitatively analyze generated responses. In Section C, we provide additional analysis of our merging method. In Section D, we discuss the limitations and the societal impacts of our work.

A Experiment Settings

A.1 Dataset Details

In this section, we provide detailed explanations of datasets used in our experiments. For all datasets, we leverage the splits that are officially provided in original repositories.

- **BindingDB** is a drug-target interaction dataset derived from experimentally measured small molecule-protein binding data, typically filtered to human proteins, and used as a binary classification benchmark with 11,054 samples for predicting whether a drug-target pair interacts.
- **BioSNAP** is a drug-target interaction dataset derived from known associations between US-marketed drugs and their human protein targets, and is used as a binary classification benchmark with 6,058 samples for predicting whether a drug-target pair interacts.
- **Human** is a drug-target interaction dataset consisting of drug-human protein pairs with highly credible negative samples and used as a binary classification benchmark with 1,375 samples to predict whether a drug-target pair interacts.
- **DrugComb** is a drug combination-cell line interaction dataset derived from standardized and harmonized drug combination screening studies across various cancer cell lines, and is used as a binary classification benchmark with 3,631 samples for predicting whether a combination of two drugs produces a synergistic or antagonistic anticancer effect in a given cancer cell line.
- **GDSC2** is a drug-cell line interaction dataset derived from the Genomics of Drug Sensitivity in Cancer project, which screens over 1,000 genetically characterized human cancer cell lines with a wide range of anti-cancer therapeutics using a newer cell screening platform introduced in 2015, and is used as a binary classification benchmark with 843 samples for predicting whether a given cancer cell line is sensitive or resistant to a specific drug.
- **CYP1A2**, **CYP2C19**, **CYP2C9**, **CYP2D6**, and **CYP3A4 Inhibition** are binary classification benchmarks for predicting whether a given drug inhibits a specific cytochrome P450 enzyme isoform involved in drug metabolism, consisting of 2,516, 2,533, 2,418, 2,626, and 2,466 samples, respectively.
- **CYP2C9**, **CYP2D6**, and **CYP3A4 Substrate** are three binary classification benchmarks to predict whether a given drug is a substrate of a specific cytochrome P450 enzyme isoform involved in drug metabolism with 134, 133, and 134 samples, respectively.
- **PAMPA** is a molecule permeability dataset derived from parallel artificial membrane permeability assay measurements, and is used as a binary classification benchmark with 407 samples for predicting whether a given molecule exhibits high or low passive permeability across an artificial membrane.
- **Immuno-Virus** is a viral protein immunogenicity dataset compiled by Li et al. [35], derived from curated viral antigen records with experimentally annotated immunogenicity labels, and is used as a binary classification benchmark with 397 samples for predicting whether a given viral protein is a protective antigen or a non-protective antigen from its amino acid sequence.
- **PBMC** is a single-cell gene expression dataset derived from peripheral blood mononuclear cells profiled by single-cell RNA sequencing, and is used as a multi-class classification benchmark with 2,398 samples for predicting the cell type of a given cell from its ranked gene-expression profile.
- **MUSIC-AVQA** [32] is an audio-visual question answering benchmark over musical performance videos, requiring joint reasoning across visual, audio, and language modalities to answer questions about sounding objects, their attributes, and temporal-spatial associations, consisting of 9,124 samples. The questions encompass both binary and open-ended formats.

A.2 Baseline Details

We evaluate our method on both the biological and general domains. For each domain, we adopt a distinct set of modality-specialized base models, while sharing six model merging baselines and three ablation variants of our proposed method.

Specialized Models for Biological Domain

- **LLaMA-3.1-8B-Instruct** [18] is the shared base large language model for all three biological specialist models, providing the general-purpose instruction-following capabilities that are subsequently adapted via LoRA for each modality-specific task.
- **Mol-LLaMA** [29] is a large molecular language model trained via multi-modal instruction tuning that integrates complementary 2D and 3D molecular encoders through a blending module, with a Q-Former projector and a LLaMA backbone fine-tuned with LoRA, providing general molecular understanding with explainability and reasoning capabilities across diverse molecular tasks.
- **Prot2Text-V2** [17] is a multi-modal sequence-to-text model that combines an ESM2-3B [36] protein sequence encoder with a LLaMA-3.1-8B-Instruct decoder via a nonlinear modality projector, using hybrid sequence-level contrastive alignment learning and instruction-based LoRA fine-tuning to generate rich functional descriptions of proteins directly from amino acid sequences.
- **Cell-o1** [16] is a cell line-specialized model fine-tuned on transcriptomic omics data, capable of handling cell line-related drug response tasks but limited to single-modality inference without understanding molecule or protein.
- **Average Merging + Task-Specific Finetune** constructs a base model by averaging the LoRA weights of all three specialist models, and subsequently fine-tunes the LoRA parameters independently on each evaluation task by sampling 2,000 examples from the training dataset for 10 epochs with a batch size of 16.

Specialized Models for General Domain

- **Vicuna-7B-v1.5** [8] is the shared base large language model upon which all three general-domain specialist models are built, adapted via LoRA for each modality-specific task.
- **Audio LLM** is a multi-modal model that combines a BEATs [7] audio encoder with a Vicuna-7B-v1.5 backbone via a Q-Former audio projector, and leverages LoRA-based fine-tuning that enables the model to capture acoustic information from speech, music, and environmental sounds.
- **Video LLM** is a multi-modal model that integrates a LanguageBind-Video [67] video encoder with a Vicuna-7B-v1.5 backbone via a two-layer MLP projector with GELU [21] activation, and is fine-tuned with LoRA to perform temporal reasoning over short video clips.
- **Image LLM** is a multi-modal model that integrates a CLIP-ViT-L/14-336 [45] image encoder with a Vicuna-7B-v1.5 backbone via a two-layer MLP projector with GELU activation, enabling visual reasoning over single images through LoRA-based instruction tuning.

Common Model Merging Methods

- **Average Merging** directly averages the parameter elements of all specialist MLLMs parameters without any additional computation.
- **TIES-Merging** [61] resolves interference among task vectors through three steps, trim, elect sign, and disjoint merge, which prune low-magnitude parameters, resolve sign conflicts by selecting the dominant direction, and merge only the parameter-aligned subset.
- **AdaMerging** [62] learns merging coefficients for task vectors in a test-time adaptation manner, using the minimization of model output entropy on unlabeled test samples as a surrogate objective without relying on original training data. In our experiments, we adopt the layer-wise AdaMerging variant which independently learns a merging coefficient for each layer of every task vector, showing better performance compared to the task-wise AdaMerging.
- **EMR-Merging** [25] is a tuning-free method that first elects a unified task vector by selecting the maximum absolute value of each parameter along the dominant sign direction, and then generates lightweight task-specific masks and rescalers to align the direction and magnitude of the unified model with each original specialist model at inference time.

- **PCB-Merging** [12] is a training-free method that constructs a parameter competition balancing matrix through intra-balancing, which measures parameter significance within individual tasks, and inter-balancing, which assesses parameter similarity across tasks, then drops low-scoring parameters and rescales the remaining ones to form the final merged model.
- **Consensus-Merging** [52] identifies and removes two classes of detrimental parameters, selfish weights that are critical exclusively to a single task and catastrophic weights that are irrelevant to all tasks, and retains only the consensus parameters that contribute positively to multi-task fusion.
- **UQ-Merge** [44] measures each model's prediction confidence via input perturbation-based uncertainty quantification, and incrementally merges models sorted in descending order of uncertainty, stopping when the merged model's uncertainty no longer decreases, thereby excluding over-confident models from the merging process.
- **RobustMerge** [66] prunes low-magnitude parameters, adaptively amplifies small singular values through complementary scaling coefficients, and applies cross-task normalization, thus maintaining direction robustness in the low-rank space to mitigate parameter interference in a parameter-efficient manner.
- **LS-Merge** [48] encodes model weights into a latent space via a transformer-based VAE, performs merging through linear interpolation in that space, and decodes back to parameters.
- **Layer-wise ES-Merging (Ours)** applies only the layer-wise global coefficient $\alpha^l_{m_j}$, derived from SWD-based embedding distribution shifts between the base and specialized models, as the merging coefficient.
- **Element-wise ES-Merging (Ours)** applies only the element-wise local coefficient $\beta^{l,n}_{m_j}$, derived from gradient-based parameter sensitivity scores with respect to fine-grained embedding distances, as the merging coefficient.
- **ES-Merging (Ours)** integrates both the global layer coefficient $\alpha^l_{m_i}$ and the local element coefficient $\beta^{l,n}_{m_i}$ into a unified merging coefficient $\lambda^{l,n}_{m_i}$ by multiplication and renormalization of elements in modalities.

A.3 Implementation Details

LoRA Configuration All specialized models adopt LoRA [23] on the self-attention projection matrices ($\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V, \mathbf{W}_O$) and MLP projection layers (gate, up, and down projections) of every transformer block, with rank $r = 8$ and scaling factor $\alpha = 32$ for the biological specialized models, and rank $r = 128$ and scaling factor $\alpha = 256$ for the general-domain specialized models.

Details of ES-Merging To construct the probe inputs, we randomly sample the 110 samples for each modality from a collection of train sets, constructing 330 samples in total. For the layer-wise merging coefficient computation, we leverage SWD with slice projection dim=1,024 and distance order p=2.0, then normalized via softmax with temperature τ=0.5. For the element-wise merging coefficient computation, the temperature τ of the softmax is predefined as 0.5.

Existing Merging Methods For all baseline methods, we follow the experimental settings reported in their respective original papers and official implementations. For LS-Merge [48], we train VAE with sequence length 16,384, batch size 512, learning rate 3×10^{-4} with 500 step warmup and cosine decay, and KL weight 10^{-4} with adaptive adjustment toward a target KL of 50, for 1,000 epochs. At merge time, model parameters are split into chunks of size 16,384, encoded into the latent space via the VAE encoder, merged via uniform-weighted Euclidean mean of the posterior means μ, and decoded back to parameter space.

LLM Inference To ensure reproducibility and reduce decoding variance, we use deterministic greedy decoding for all methods, without sampling. Specifically, at each decoding step, the model selects the token with the highest predicted probability.

Compute Resources All experiments are conducted on NVIDIA B200 and H200. For the computation of merging coefficients, it takes one and a half hours, while the experiments on the instance-varying interaction prediction and the target-fixed functionality prediction take 25 and 7 hours.

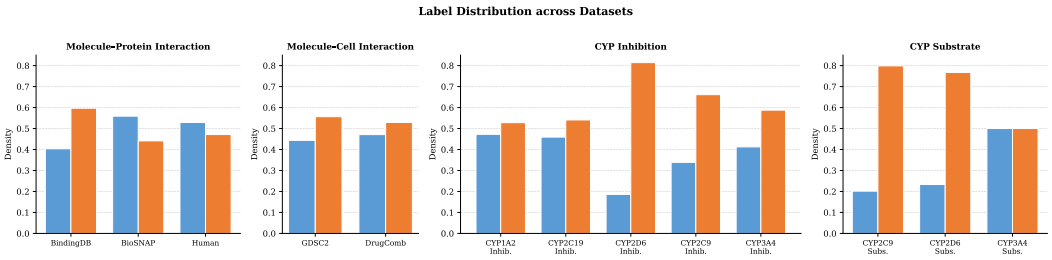


Figure 7: Label distribution across datasets. Each panel shows positive and negative sample counts for Molecule-Protein Interaction, CYP Inhibition, CYP Substrate, and Molecule-Cell Interaction tasks.

A.4 Evaluation Prompts

To impart the task-specific knowledge, we provide 5-shot examples from the training set with the prompt templates in Table 19 and Table 20. The retrieval strategy is designed per task as follows:

- **Molecule-Protein Interaction:** Training samples whose target protein sequence exactly matches the query are first collected. If more than five exact matches exist, the top-5 are selected by Tanimoto similarity [57] of Morgan fingerprints [47] between their molecules and the query molecule. If fewer than five are found, the remaining slots are filled with samples from proteins with the highest cosine similarity of protein embeddings from ESM2 to the query protein. Among candidates from the same similar protein, the one with the highest Tanimoto similarity to the query molecule is preferred.
- **Drug-Cell Interaction:** Training samples whose cell line shares an exact match on the top-50 expressed gene set are priorly collected. If fewer than five are found, additional samples are drawn from cell lines with the highest Jaccard similarity on gene sets. When multiple candidates exist from the same cell line, the one with the highest Tanimoto similarity to the query molecule is selected. For DrugComb, molecule similarity is computed as the average Tanimoto similarity across both drugs.
- **CYP Inhibition/Substrate:** Since all samples share the same CYP enzyme target, protein-level filtering is unnecessary. The top-5 examples are selected solely by Tanimoto similarity between the training molecules and the query molecule.
- **PAMPA/Immuno-Virus/PBMC:** As these single-modality tasks correspond to out-of-distribution datasets that the specialist models have not been trained on, we employ a task-grounded zero-shot prompt that consists of (i) a concise information of the task, and (ii) explanations of the analytical factors that the model should consider when reasoning toward the answer. The model is then instructed to reason step-by-step over these factors before producing the final answer.
- **MUSIC-AVQA:** Following Chen et al. [5], we issue the same questions across multiple input-modality configurations of V, I, A, $V{+}I$, $V{+}A$, $I{+}A$, and $V{+}I{+}A$, where V, I, and A denote video, image, and audio inputs, respectively. For each configuration, only the corresponding modalities are inserted into the prompt while the question text and answer format remain identical.
- **AVQA:** We evaluate four-option multiple-choice questions about real-world scenes, objects, actions, and sound-producing events [63] using the same seven input-modality configurations as MUSIC-AVQA. Each question has one correct answer and three distractors. For each configuration, the merged model receives the question together with only the selected video, image, and/or audio inputs.

A.5 Evaluation Metric

As shown in Figure 7, while some datasets exhibit a relatively balanced distribution between positive and negative samples, others such as BindingDB and CYP2D6 Inhibition display a pronounced class imbalance. In particular, the CYP Substrate task contains only approximately 130 samples per dataset, posing a critical challenge of absolute data scarcity. In such imbalanced and low resource settings, relying solely on accuracy as an evaluation metric can be misleading, as a model that predominantly

predicts the majority class may still achieve inflated scores without genuinely capturing minority-class patterns. Therefore, we leverage the macro-F1 that addresses this limitation by computing the F1 score for each class independently and averaging them with equal weight regardless of class frequency, thereby providing a fair assessment of predictive performance across all classes.

B Qualitative Analysis

In this section, we qualitatively compare the generated responses of each task-specific fine-tuned model with those of ES-Merging, as shown in Table 8 and Table 9.

B.1 Molecule-Protein Interaction Prediction

Table 8 presents a generated response on the molecule-protein interaction prediction task from the BioSNAP dataset, where the molecule is paramethasone and the target protein is the glucocorticoid receptor. Although both ES-Merging and the task-specific fine-tuning model correctly predict the label, the two responses differ substantially in quality. ES-Merging correctly identifies the target as the glucocorticoid receptor, a nuclear receptor that regulates gene expression in response to glucocorticoids, and recognizes the molecule as a fluorinated steroid with glucocorticoid properties. Based on these complementary observations, it reasons that the molecule is likely to interact with the glucocorticoid receptor, which is consistent with paramethasone being annotated as a glucocorticoid receptor agonist targeting NR3C1. This reasoning process suggests that ES-Merging successfully integrates chemical structure-level knowledge acquired from the molecule expert model with biological function-level knowledge acquired from the protein expert model through embedding-space-signal-based merging. In contrast, the task-specific fine-tuned model outputs only the label "Interacts" without providing any biological or chemical rationale. Because it was trained on instruction data containing labels only, its predictions are less interpretable, whereas ES-Merging integrates modality-expert knowledge to produce interpretable reasoning even for unseen cross-modal tasks.

B.2 Molecule-Cell Interaction Prediction

Table 9 presents a qualitative analysis of the Drug-Cell Interaction prediction task from the GDSC2 dataset, where the drug is Gefitinib, and the cell line is OVCA420. ES-Merging first identifies the given molecule as a quinazoline derivative and correctly recognizes it as a potent inhibitor of the epidermal growth factor receptor (EGFR) tyrosine kinase. It also accurately identifies the given cell line as a cancer cell, then focuses on genes such as RPS6, EIF1, and GNB2L1 in the ranked gene-expression list to interpret the cell as actively proliferating and potentially dependent on EGFR signaling. On the basis of this, ES-Merging correctly recognizes that the drug's mechanism of action is effective against the given cell. This reasoning chain spanning from molecular substructure to drug class, target pathway, cell type identification, and cell-level interpretation demonstrates that the structural knowledge from the molecule expert model and the transcriptomic knowledge from the cell expert model are functionally integrated through ES-Merging, while the model with task-specific finetuning again outputs only the label "Sensitive" without any biological rationale.

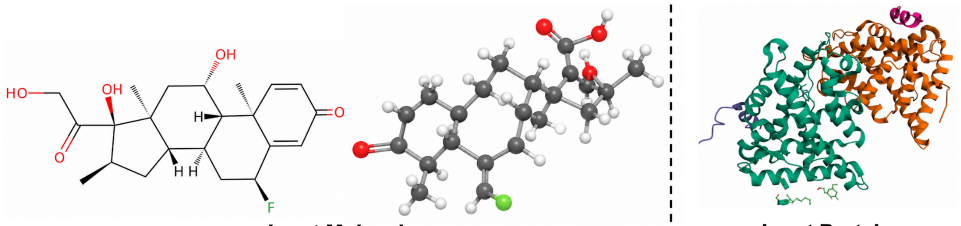
C Further Analyses

We provide further analyses regarding the merging coefficients, single-modal tasks, ablation study, temperature parameter τ, probe prompt type, probe input sensitivity, embedding visualizations, and additional AVQA evaluation.

C.1 Merging Coefficients

C.1.1 Layer-wise Merging Coefficients

Projection Dimension Size of SWD When computing SWD for the layer-wise merging coefficients, we vary projection dimension sizes on representative datasets for each task, and derived merging coefficients accordingly. As shown in Table 10, performance consistently improves as the projection size increases, with 1024 dimensions achieving the best results. This is because a larger number of projections enables a more precise approximation of the distributional differences in high-dimensional



Molecule Name: Paramethasone

SMILES: C[C@@H]1C[C@H]2[C@@H]3C[C@@H](C4=CC(=O)C=C[C@@]4([C@H]3[C@H](C[C@@]2([C@]1(C(=O)CO)O)C)O)C)F

Protein: Glucocorticoid receptor

Protein Sequence: MDSKESLTPGREENPSSVLAQERGDVMDFYKTLRGGATVKVSASSPSLAVA SQSDSKQRRLLVDFPKGSVSNAQQPDLSKAVSLSMGLYMGETETKVMGNDLGFPQQGQISLSS GETDLKLLEESIANLNRSTSVPENPKSSASTAVSAAPTEKEFPKTHSDVSSEQQHLKGQTGTNGG NVKLYTTDQSTFDILQDLEFSSGSPGKETNESPWRSDLLIDENCLLSPLAGEDDSFLLEGNSNED CKPLILPDTKPKIKDNGDLVLSSPSNVTLPQVKTEKEDFIELCTPGVIKQEKLGTVYCQASFPGA NIIGNKMSAISVHGVSTSGGQMYHYDMNTASLSQQQDQKPIFNVIPPIPVGSENWNRCQGSGDD NLTSLGTLNFPGRTVFSNGYSSPSMRPDVSSPPSSSSTATTGPPPKLCLVCSDEASGCHYGVLTC GSCKVFFKRAVEGQHNYLCAGRNDCIIDKIRRKNCPACRYRKCLQAGMNLEARKTKKKIKGIQ QATTGVSQETSENPGNKTIVPATLPQLTPTLVSLLEVIEPEVLYAGYDSSVPDSTWRIMTTLNML GGRQVIAAVKWAKAIPGFRNLHLDDQMTLLQYSWMFLMAFALGWRSYRQSSANLLCFAPDLI INEQRMTLPCMYDQCKHMLYVSSELHRLQVSYEEYLCMKTLLLLSSVPKDGLKSQELFDEIRM TYIKELGKAIVKREGNSSQNWQRFYQLTKLLDSMHEVVENLLNYCFQTFLDKTMSIEFPEMLA EIITNQIPKYSNGNIKKLLFHQK

User: Determine whether the given molecule interacts with the protein by following the example.

Ground Truth: Interacts

ES-Merging (Ours)

To determine whether the given molecule interacts with the protein, we need to analyze the molecular structure and properties of both the protein and the molecule.

The protein is a glucocorticoid receptor, which is a type of nuclear receptor that plays a crucial role in regulating gene expression in response to glucocorticoids. The protein has a specific binding site for glucocorticoids, and it is activated by the binding of these hormones.

The molecule is a synthetic glucocorticoid, which is a type of steroid hormone that is designed to mimic the effects of natural glucocorticoids. It has a similar molecular structure to natural glucocorticoids and is known to bind to the glucocorticoid receptor.

Based on the molecular structure and properties of both the protein and the molecule, it is likely that the molecule interacts with the protein. The molecule is designed to bind to the glucocorticoid receptor, and it has a similar structure to natural glucocorticoids, which are known to interact with this protein.

Therefore, the final answer is:

Final answer: Interacts

Models with Task-Specific Finetuning

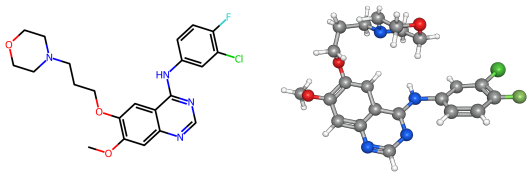
Final answer: Interacts

Table 8: Case study on Molecule-Protein Interaction prediction.

space, leading to more accurate computation of the layer-wise merging coefficients. Based on this finding, we set the SWD projection size to 1024 in our main experiments.

C.1.2 Element-wise Merging Coefficients

The coefficient patterns differ across q/k/v/o_proj modules even within the same layer. As shown in Figure 10, the q/k/v projection modules and the o projection module emphasize different elements even at Layer 0. Within the same module, the coefficient distributions between LoRA A and LoRA B



Drug Name: Gefitinib

SMILES: COC1=C(C=C2C(=C1)N=CN=C2NC3=CC(=C(C=C3)F)Cl)OCCCN4CCOCC4

Cell Line: OVCA420

Top 50 Genes: RPL35, FTL, MT-ND4, GAPDH, AKR1B10, UBC, RPS6, RPL24, EIF1, GNB2L1, SPP1, RPS11, AKR1C1, RPL19, GNG5, AKR1B10P1, ACTG1, MT-CO3, RPL5, FAU, RPS15A, TXN, RPL8, MIF, LDHB, KRT19, EIF4H, RPL11, ALDH1A1, MGST1, RPS12, CLIC1, SNRPD2, RPL30, H2AFX, NQO1, PRDX1, RPS5, RPLP1, H2AFZ, TUBB4B, FAUP1, NACA, RPL14, HIST1H2BK, MYL12B, SHFM1P1, HINT1P1, RPL27, HIST1H4C

User: Determine whether the given cell is sensitive or resistant to the given molecule based on the ranked gene-expression list.

Ground Truth: Sensitive

ES-Merging (Ours)

The given molecule is 4-(3-fluoro-4-methoxyphenyl)-7-methyl-6-oxo-1,4-dihydroquinazoline-3-carboximidamide, which is a quinazoline derivative. This type of molecule is known to be a potent inhibitor of the epidermal growth factor receptor (EGFR) tyrosine kinase. The cell line provided is a cancer cell, and the presence of genes such as RPS6, EIF1, and GNB2L1 in the list suggests that the cell is actively proliferating and may be dependent on EGFR signaling for growth and survival. Given the presence of these genes and the fact that the molecule is an EGFR inhibitor, it is likely that the cell is sensitive to the molecule.

Final answer: Sensitive

Models with Task-Specific Finetuning

Final answer: Sensitive

Table 9: Case study on Molecule-Cell Interaction prediction.

Size	Human	GDSC2	CYP2C9 Inh.	CYP2C9 Sub.
32	60.7	90.6	68.2	53.7
256	62.0	93.1	68.9	48.5
1024	**62.0**	**94.1**	**72.5**	**64.2**

Table 10: Additional studies on projection size of SWD for ES-Merging across Molecule-Protein and Molecule-Cell Interaction tasks.

exhibit distinct patterns. In particular, at Layer 0 of the q projection module, LoRA A shows relatively balanced coefficients across all modalities, whereas LoRA B contains regions where molecule and protein are more prominent. On the other hand, coefficient distributions are far from uniform, indicating that layer-wise global coefficients alone cannot capture the fine-grained specialization differences and supporting the necessity of element-wise local coefficients.

C.1.3 Final Coefficients of ES-Merging

The layer-wise global coefficient $\alpha^l_{m_i}$ and the element-wise local coefficient $\beta^{l,n}_{m_i}$ focus on different regions. The layer-wise coefficient captures coarse-grained distributional shifts in the embedding space, while the element-wise coefficient reflects fine-grained importance at the individual parameter

level. As discussed in Section 4.4, by multiplying these two coefficients and normalizing, regions where both coefficients assign high importance are amplified, whereas regions where only one side is high and the other is low are suppressed. This enables merging that simultaneously incorporates global layer-level specialization signals and local element-level specialization signals. As shown in Figure 11, compared to the element-wise-only coefficient distributions, the combined results demonstrate that the overall scale of coefficients is adjusted according to the global importance of each layer, while the fine-grained element-level patterns are preserved.

Table 11: Accuracy (%) on CYP2D6 inhibition with different modality aggregation strategies for coefficient estimation. Each column uses only the indicated coefficient type.

Scoring strategy	**Layer-wise only**	**Element-wise only**
All modalities	79.2	77.6
Matched modality only	78.3	71.2

C.1.4 Modality Aggregation for Coefficient Estimation

We compare two strategies for aggregating the modality-conditioned signals used to estimate merging coefficients. Our default strategy equally aggregates signals from molecule, protein, and cell tokens for each expert. The matched-modality strategy uses only signals from the modality corresponding to each expert, e.g., molecule-token signals for the molecule expert. We evaluate both strategies separately in the layer-wise-only and element-wise-only variants on CYP2D6 inhibition. As shown in Table 11, relying only on the matched-modality signal underperforms all-modality aggregation, especially in the element-wise setting. This suggests that a single modality signal is insufficient for stable fine-grained attribution and supports jointly considering responses across all modalities to capture complementary contribution signals beyond modality-specific scoring.

	Molecule	**Protein**	**Cell**
Mol-LLaMA [29]	75.68	88.02	0.00
Prot2Text-V2 [17]	39.80	85.25	0.00
Cell-o1 [16]	60.89	77.81	77.84

Table 12: Performance comparison of different modality-specialized LLMs, evaluating their knowledge and capabilities not only on tasks within their target modality but also across other modalities. We report accuracy across each subset.

C.2 Detailed Analysis of Single-modal Knowledge Preservation

Table 12 reports whether each modality-specialized LLM retains a certain level of knowledge not only for its target modality but also for tasks in other modalities. We observe that Mol-LLaMA and Prot2Text-V2 attain zero accuracy on Cell tasks, whereas all specialized models possess a certain degree of cross-modal knowledge on the Molecule and Protein tasks. This distribution directly explains the results in Table 3. The drop in Cell performance compared to Cell-o1 alone arises because merging three models inevitably incurs a certain amount of information loss when the target model is combined with others that hold little to no knowledge of the corresponding task. Despite this constraint, ES-Merging preserves Cell knowledge most effectively among all merging methods and outperforms every baseline by a clear margin. In contrast, for the Molecule and Protein tasks, the non-target specialized models also encode a certain level of knowledge about each other's modalities, allowing ES-Merging to effectively integrate these complementary signals and even surpass the original modality-specialized MLLMs. These results confirm that ES-Merging selectively preserves and amplifies modality-relevant knowledge when it is present across the source models, while still retaining as much of that knowledge as possible even when it resides in only a subset of them.

	Molecule-Protein Interaction				Molecule-Cell Interaction			CYP Inhibition						CYP Substrate			
Coefficient Type	BindingDB	BioSNAP	Human	*Avg.*	DrugComb	GDSC2	*Avg.*	CYP1A2	CYP2C19	CYP2C9	CYP2D6	CYP3A4	*Avg.*	CYP2C9	CYP2D6	CYP3A4	*Avg.*
Layer-wise	65.5	68.2	57.0	63.6	80.1	90.2	85.2	76.3	**71.4**	72.3	79.2	70.2	73.9	61.2	54.9	55.2	57.1
Element-wise	**66.5**	66.6	61.4	64.9	79.3	**94.1**	86.7	76.0	69.1	70.7	77.6	69.8	72.7	**65.7**	57.6	58.2	60.5
Layer × Element Mixed	66.0	**69.1**	**62.0**	**65.7**	**80.7**	**94.1**	**87.4**	**77.4**	70.6	**72.5**	**80.7**	**71.3**	**74.5**	64.2	**60.9**	**60.5**	**61.9**

Table 13: Full results of the ablation studies on merging coefficient. We report accuracy across all datasets and their averages. **Bold** indicates the best and underline indicates the second best.

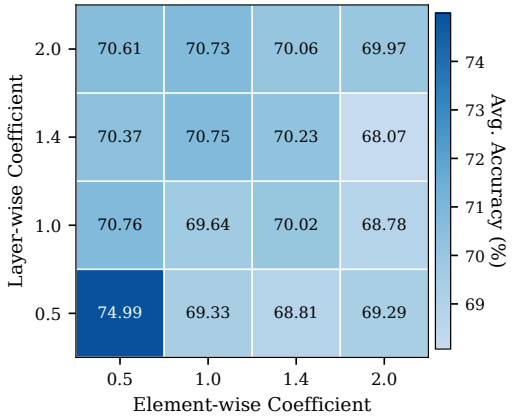


Figure 8: Performance with diverse temperatures τ for the layer-wise and element-wise coefficients in ES-Merging. Each cell reports the average accuracy (%) across CYP2D6 Inhibition, CYP2D6 Substrate, Human, and GDSC2.

C.3 Details of Ablation Study

Table 13 reports the detailed per-dataset results corresponding to the averages of each task group in Table 5. The layer-wise ES-Merging and element-wise ES-Merging each dominate on different datasets. Despite this dataset-level variations, Layer×Element ES-Merging consistently achieves the best or comparable performance across all individual datasets. These results confirm that the two coefficients capture specialization at different granularities depending on task and dataset characteristics, and their combination compensates for the weaknesses of either side, yielding consistent and robust merging performance at the individual dataset level as well.

C.4 Effect of Temperature parameter τ for Controlling Softmax

The temperature parameter τ controls the sharpness of the softmax normalization used to convert embedding-space importance scores into merging coefficients. To see the effect of τ, we compare the performance over $\tau_{\text{layer}}, \tau_{\text{elem}} \in \{0.5, 1.0, 1.4, 2.0\}$ by the average accuracy across four representative datasets: CYP2D6 Inhibition, CYP2D6 Substrate, Human, and GDSC2. As shown in Figure 8, the best performance is achieved when both temperatures are set to 0.5. Increasing either temperature generally degrades performance, indicating that overly smooth coefficient distributions weaken the modality-specific signals captured from the embedding space.

C.5 Probe Prompt Type

Probe Prompt Settings To compute the embedding-signal based merging coefficients, we feed a probe prompt that contains placeholder tokens for each modality into each expert model and extract their hidden representations. We consider two variants of the probe prompt. In the *concatenated prompt* setting, the placeholders for all three modalities are packed into a single input sequence, so that the three modality signals are exposed to every expert model within the same context (Figure 3). In contrast, in the *separated prompt* setting, we construct three modality-specific prompts so that each expert model processes inputs in which the modalities are kept isolated, and the coefficients are computed from these per-modality representations.

Table 14: Full results of the additional studies on probe prompt types for cross-modality tasks. We report accuracy across all datasets and their average per task group.

	Molecule-Protein Interaction				Molecule-Cell Interaction			CYP Inhibition						CYP Substrate			
Coefficient Type	BindingDB	BioSNAP	Human	*Avg.*	DrugComb	GDSC2	*Avg.*	CYP1A2	CYP2C19	CYP2C9	CYP2D6	CYP3A4	*Avg.*	CYP2C9	CYP2D6	CYP3A4	*Avg.*
Separated Prompt	65.3	67.6	61.9	64.9	77.1	93.9	85.5	76.5	**71.1**	71.2	79.0	69.6	73.5	59.4	57.1	61.9	59.5
Concatenated Prompt	**66.0**	**69.1**	**62.0**	**65.7**	**80.7**	**94.1**	**87.4**	**77.4**	70.6	**72.5**	**80.7**	**71.3**	**74.5**	64.2	**60.9**	**60.5**	**61.9**

Table 15: Full results of the additional studies on probe prompt types for single-modality tasks. We report accuracy and their average accuracy over all datasets.

	Molecule	Protein	Cell	*Avg.*
Separated Prompt	**81.57**	80.83	52.77	71.72
Concatenated Prompt	77.64	**86.51**	**61.49**	**75.21**

Analysis As summarized in Table 14 and Table 15, the concatenated prompt consistently outperforms the separated prompt across nearly all tasks. Not only on cross-modality tasks but also on single-modality tasks, the concatenated prompt achieves higher average accuracy than the separated prompt, with particularly large gains on the Protein and Cell modalities. This suggests that exposing all modality placeholders within a shared context allows each expert to produce coefficients that already encode cross-modal interaction, leading to more discriminative merging signals. In contrast, representations obtained from modality-isolated prompts lack the contrastive information across modalities, which in turn limits the modality-specific discriminativeness that the derived coefficients can provide. Based on this analysis, we adopt the concatenated prompt as our default probe setting in all main experiments.

C.6 Probe Input Sensitivity

We further validate the robustness of ES-Merging with respect to probe input construction. Specifically, we vary the size of probe inputs and compare the results with those obtained using another probe input set. As shown in Table 16, ES-Merging with different probe input sizes generally outperforms the parameter-signal-based merging methods, demonstrating its robustness to the probe input size. Interestingly, increasing the probe input size does not bring a significant performance gain, while using a smaller size degrades the performance. This suggests that a size of 330 provides a favorable trade-off between performance and computational cost. On the other hand, the original probe input set shows a marginal performance gap compared to another set of the same size, indicating that ES-Merging is insensitive to the choice of probe inputs.

Table 16: Additional studies of probe dataset size and sampling for ES-Merging across Molecule-Protein and Molecule-Cell Interaction tasks.

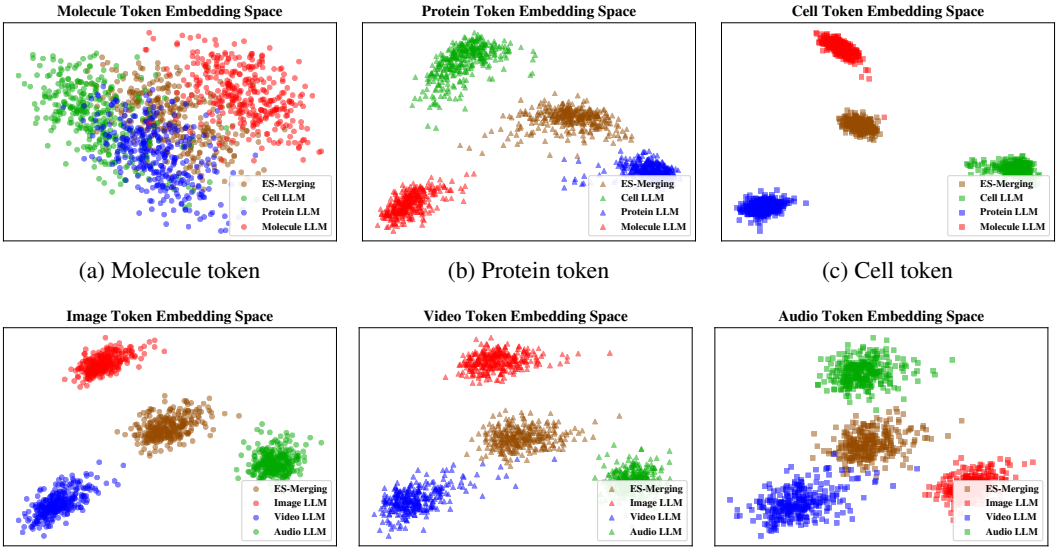
Size	**Human**	**GDSC2**	**CYP2D6 Inh.**	**CYP2D6 Sub.**
50	61.0	91.2	78.7	59.4
600	61.2	92.4	79.1	60.9
330 (Another subset)	61.8	93.9	79.6	60.5
330	62.0	94.1	80.7	60.9

Table 17: Mean accuracy (%) and standard deviation of ES-Merging over three runs with independently sampled probe sets. The reported variability is due to probe sampling.

Dataset	**Mean $\pm$ Std.**
Human	61.8 ± 0.3
GDSC2	93.9 ± 0.2
CYP2D6 Inhibition	80.2 ± 0.6
CYP2D6 Substrate	60.8 ± 0.2

Stability across Probe-Sampling Seeds As we use greedy decoding with temperature 0 and deterministic similarity-based few-shot retrieval, repeated inference with the same merged model and inputs produces identical predictions. Randomness arises when selecting probe inputs used to estimate the merging coefficients. To quantify this variability, we repeat ES-Merging three times using independently sampled probe sets and report the mean accuracy and standard deviation across runs in Table 17. The standard deviations range from only 0.2 to 0.6 percentage points across the four evaluated datasets, indicating that ES-Merging remains stable under variations in probe sampling.

C.7 Embedding Visualization



(a) Molecule token (b) Protein token (c) Cell token

(d) Image token (e) Video token (f) Audio token

Figure 9: Embedding visualization of the last transformer block for each specialized LLM and the merged model with our method for (a) molecule, (b) protein, and (c) cell, (d) image, (e) video, and (f) audio tokens.

Figure 9 visualizes the embedding distributions of each specialist model and the model merged by ES-Merging at the last transformer block for each modality token. In each modality token space, the specialist models form distinct distributions, indicating that each model has learned modality-specific representations. The model merged by ES-Merging is positioned between the specialist distributions without being biased toward any particular specialist, while being relatively close to the distribution of the specialist corresponding to each modality token. This suggests that ES-Merging integrates the modality-specific knowledge of each specialist model in a balanced manner, while preserving the specialization for each modality. Notably, this pattern holds consistently across both biological and general domains, demonstrating that ES-Merging generalizes well across heterogeneous task settings rather than being tailored to a specific domain.

C.8 Additional Evaluation on AVQA

To further examine generalization beyond musical scenarios, we evaluate the merged video-, image-, and audio-specialized model on AVQA [63], which covers diverse real-world scenes, objects, actions, and sound-producing events. We compare with Average, TIES, EMR, and PCB merging under the same seven input-modality configurations as in MUSIC-AVQA. As shown in Table 18, ES-Merging outperforms all evaluated baselines across all seven input configurations, showing consistent advantages in both single-modality and multimodal settings. For ES-Merging, all three two-modality combinations improve over their constituent single-modality inputs. It is also the only evaluated method to achieve its highest accuracy when all three modalities are provided, reaching 79.32%. Together, these results suggest that ES-Merging can effectively use complementary modality information across diverse real-world scenes, extending its applicability beyond the musical scenarios evaluated in MUSIC-AVQA.

Table 18: Performance comparison on AVQA across input-modality configurations. We report accuracy (%); V, I, and A denote video, image, and audio, respectively. **Bold** and underline indicate the best and second best performances.

Method	V	I	A	V + I	V + A	I + A	V + I + A
Avg. Merging	62.20	73.32	58.48	66.40	67.87	75.86	69.06
TIES-Merging [61]	43.38	69.99	55.78	43.73	46.62	71.43	52.16
EMR-Merging [25]	63.90	74.83	59.12	70.86	71.82	76.05	71.18
PCB-Merging [12]	40.59	68.55	54.69	39.66	43.12	70.34	48.61
ES-Merging (Ours)	**74.93**	**75.95**	**61.11**	**78.36**	**79.00**	**77.94**	**79.32**

D Limitation and Societal Impacts

Limitation In this work, we present ES-Merging, a novel MLLM merging framework that derives layer-wise and element-wise merging coefficients by leveraging embedding-space signals. Although ES-Merging achieves strong performance across cross-modal reasoning and single-modal knowledge preservation, it still requires an additional pre-merging stage to estimate embedding-space signals from probe inputs, introducing a one-time computational overhead compared to methods such as average merging. Nevertheless, this overhead is substantially lower than that of iterative optimization-based approaches, since ES-Merging estimates the merging coefficients once rather than repeatedly updating model parameters. Future work could explore more lightweight signal estimation strategies, such as reusing cached representations or estimating sparse coefficients.

Societal Impacts We demonstrate that ES-Merging effectively integrates modality-specialized biological MLLMs by leveraging embedding space signals, enabling improved cross-modal effect prediction across diverse biological tasks. This capability has the potential to accelerate drug discovery by reducing reliance on costly and time-consuming wet-lab experiments. However, there is a risk that such capabilities could be maliciously exploited to identify harmful or toxic substances. We strongly hope that our method will be used solely for beneficial scientific purposes.

Table 19: In-context learning prompt templates for the Molecule-Protein Interaction and CYP prediction task. Angle-bracketed tokens are replaced with the corresponding encoder embeddings: `<protein>` and `<mol>` for the target pair, and `<protein k>` and `<molecule k>` for each of the k few-shot examples. The `{label k}` is replaced with the ground-truth label string for each example pair in the few-shot context.

Prompt for Molecule-Protein Interaction Prediction

System

You are an expert specialized in drug discovery and molecular biology. You will be given a protein and a molecule. Your task is to **determine whether a given molecule interacts with a specific protein.**

User

Determine whether the given molecule interacts with the protein by following the examples.

Examples:
Example 1: Protein: `<protein 1>` Molecule: `<molecule 1>` Final answer: `{label 1}`
⋮
Example k: Protein: `<protein k>` Molecule: `<molecule k>` Final answer: `{label k}`

Protein: `<protein>` Molecule: `<mol>`
Your final answer must be exactly one of: **'Final answer: Interacts' or 'Final answer: Does not interact'.**

Prompt for CYP Inhibition Prediction

System

You are an expert specialized in drug discovery and molecular biology. You will be given a protein and a molecule. Your task is to **determine whether a given molecule inhibits a specific protein.**

User

Determine whether the given molecule inhibits the protein by following the examples.

Examples:
Example 1: Protein: `<protein 1>` Molecule: `<molecule 1>` Final answer: `{label 1}`
⋮
Example k: Protein: `<protein k>` Molecule: `<molecule k>` Final answer: `{label k}`

Protein: `<protein>` Molecule: `<mol>`
Your final answer must be exactly one of: **'Final answer: Inhibit' or 'Final answer: Does not inhibit'.**

Prompt for CYP Substrate Prediction

System

You are an expert specialized in drug discovery and molecular biology. You will be given a protein and a molecule. Your task is to **determine whether a given molecule is a substrate of a specific protein.**

User

Determine whether the given molecule is a substrate of the protein by following the examples.

Examples:
Example 1: Protein: `<protein 1>` Molecule: `<molecule 1>` Final answer: `{label 1}`
⋮
Example k: Protein: `<protein k>` Molecule: `<molecule k>` Final answer: `{label k}`

Protein: `<protein>` Molecule: `<mol>`
Your final answer must be exactly one of: **'Final answer: Substrate' or 'Final answer: Not a substrate'.**

Table 20: In-context learning prompt template for the Molecule-Cell Interaction prediction task. Angle-bracketed tokens are replaced as follows: target molecule token <mol> and <molecule k> for each of the k few-shot examples are substituted with the corresponding encoder embedding, while the gene tokens <gene> are replaced with the corresponding cell line string. The {label k} is replaced with the ground-truth label string for each example pair in the few-shot context.

Prompt for Molecule-Cell Interaction Prediction: GDSC2

System

You are an expert specialized in cancer pharmacogenomics and drug response prediction. You will be given an anticancer molecule and a cancer cell represented as a list of gene names ordered by expression, where the most highly expressed genes appear first in descending order. Your task is to predict whether the molecule will biologically suppress the cancer cell. **If the molecule is expected to suppress/inhibit cancer cell growth or viability, label it Sensitive, otherwise label it Resistant.**

User

Determine whether the given cell is sensitive or resistant to the given molecule based on the ranked gene-expression list.

Examples:
Example 1: Cell: <gene 1> Molecule: <molecule 1> Final answer: {label 1}
⋮
Example k: Cell: <gene k> Molecule: <molecule k> Final answer: {label k}

Cell: <gene> Molecule: <mol>
Your final answer should be formatted as either: **'Final answer: Sensitive' or 'Final answer: Resistant'.**

Prompt for Molecule-Cell Interaction Prediction: DrugComb

System

You are an expert specialized in cancer pharmacogenomics and anticancer drug-combination response prediction. You will be given two anticancer molecules and a cancer cell represented as a ranked list of gene names (highest expression first). **Your task is to predict the binary interaction outcome in terms of anticancer effect in that cell line.**

Labels:
Synergistic – the combination produces a stronger anticancer effect than expected,
Antagonistic – the combination produces a weaker anticancer effect than expected.

User

Predict whether the drug pair is Synergistic or Antagonistic in the given cell line.

Examples:
Example 1: Cell: <gene 1> Molecule1: <molecule1 1> Molecule2: <molecule2 1>
Final answer: {label 1}
⋮
Example k: Cell: <gene k> Molecule1: <molecule k> Molecule2: <molecule2 k>
Final answer: {label k}

Cell: <gene> Molecule1: <mol1> Molecule2: <mol2>
Your final answer should be formatted as either: **'Final answer: Synergistic' or 'Final answer: Antagonistic'.**

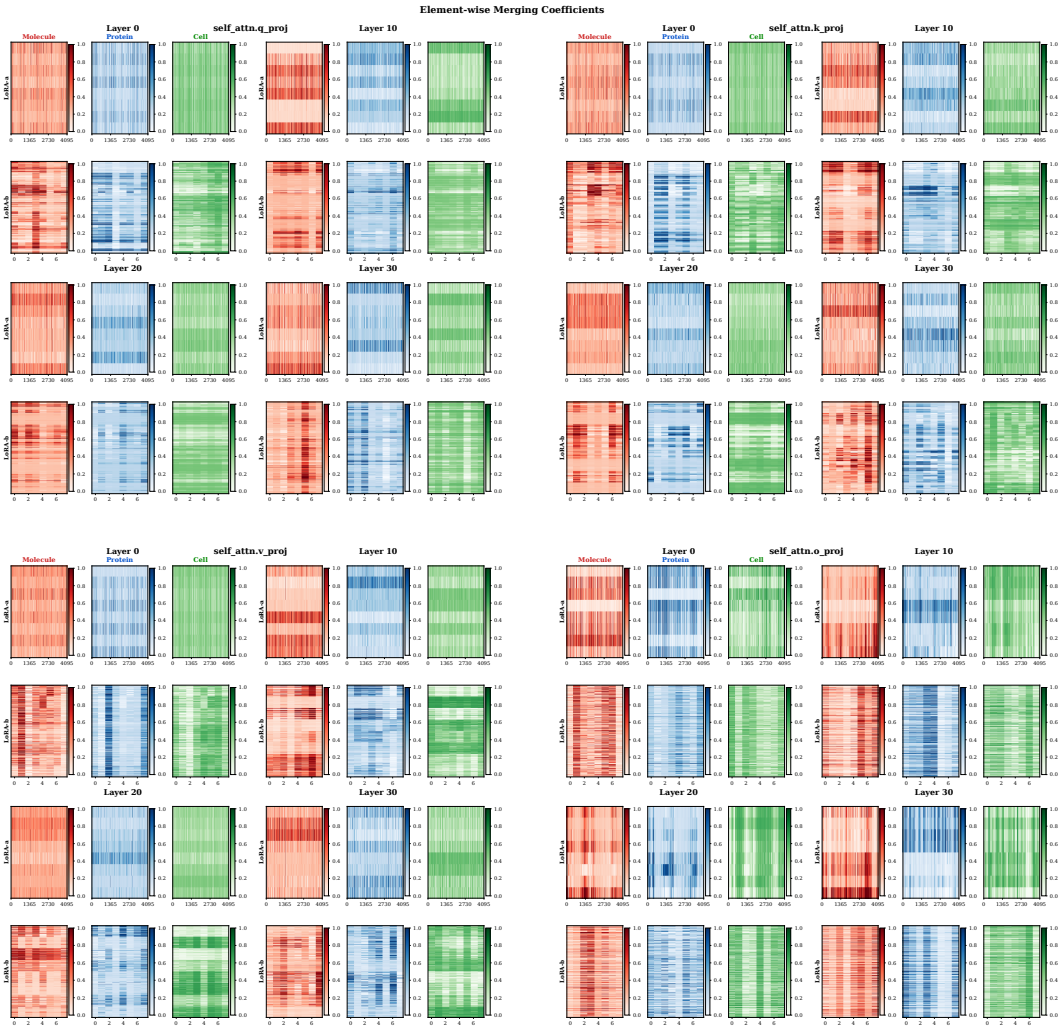
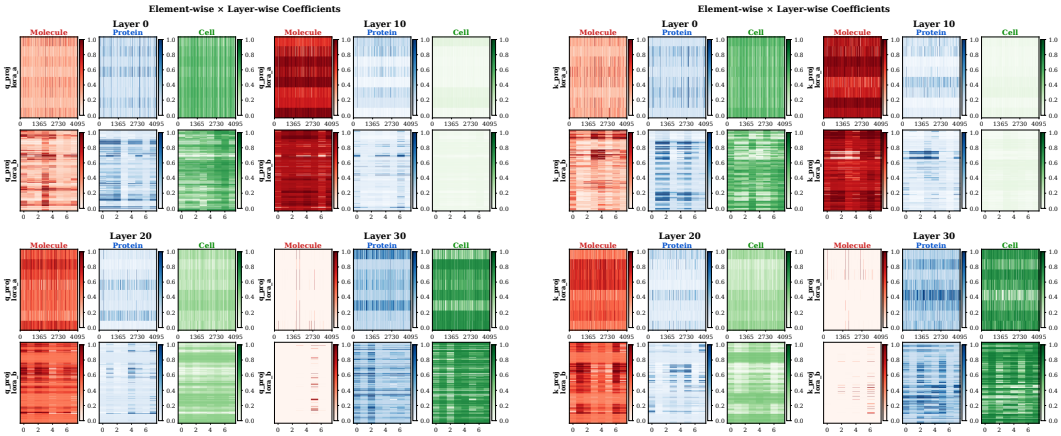
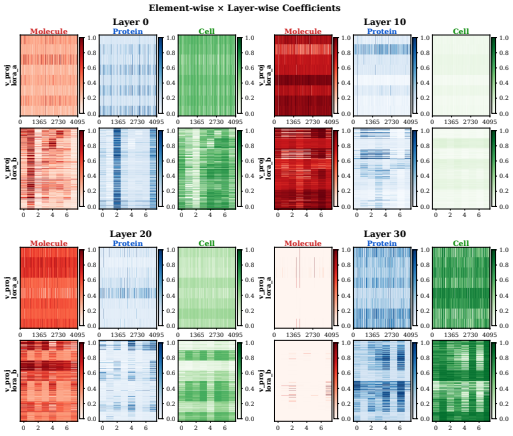


Figure 10: Element-wise merging coefficients of the LoRA parameters for `self_attn.q/k/v/o_proj` modules at layers 0, 10, 20, and 30. Rows correspond to LoRA-A and LoRA-B, and columns correspond to Molecule, Protein, and Cell modalities.

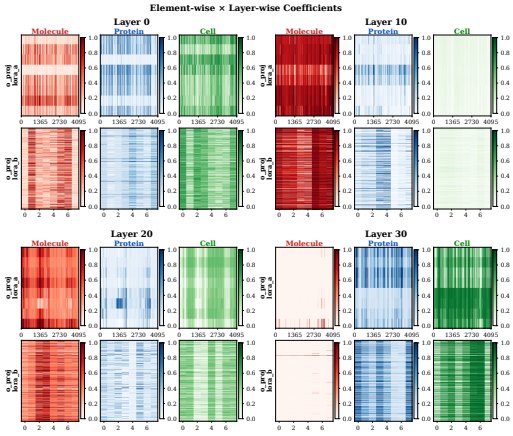


(a) q projection

(b) k projection



(c) v projection



(d) o projection

Figure 11: Element × Layer merging coefficients of the LoRA parameters for `self_attn.q/k/v/o_proj` modules at Layers 0, 10, 20, and 30, respectively.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We explain the main limitation of existing parameter-signal-based merging methods for merging the modality-specialized MLLMs and propose the embedding-signal-based merging framework (ES-Merging).

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitation is discussed in Section D.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: [N/A]

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We leveraged the greedy decoding during the LLM inference to ensure reproduction. Also, we have explained the experimental setup and the details of datasets, baselines and implementations along with used prompts in Section 5.1, A.1, A.2, A.3, and A.4, respectively.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide the code via OpenReview.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: We have explained the experimental setup and the details of datasets, baselines and implementations along with used prompts in Section 5.1, A.1, A.2, A.3, and A.4, respectively.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We report mean accuracy and standard deviation across three runs with independently sampled probe sets on four representative datasets in Appendix C.6 (Table 17). These statistics quantify variability from probe sampling; inference uses greedy decoding and deterministic few-shot retrieval.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: The compute resources are explained in Section A.3.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification:

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: The societal impacts are described in Section D.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification:

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We leverage open-sourced datasets and models.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification:

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification:

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification:

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification:

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.