Codebook-Guided Cross-Modal Knowledge Distillation for Structurally Heterogeneous Features

Dae Ung Jo
Kyungpook National University
daeung.jo@knu.ac.kr

Jongin Lim
AX/PI Center, Samsung Electronics
jonny.lim@samsung.com

YoungJoon Yoo*
Chung-Ang University, SNUAILAB
yjyoo3312@cau.ac.kr

Daeho Um*
University of Seoul
daehoum@uos.ac.kr

Abstract

Cross-modal knowledge distillation transfers knowledge from a teacher modality to a student modality. Existing feature-level alignment methods typically assume that teacher and student features reside in structurally alignable representation spaces. However, this assumption does not hold when cross-modal features are structurally heterogeneous and lack clear unit-level correspondence, such as 2D spatial visual grids and 1D temporal audio sequences, thereby limiting the applicability of feature-level alignment. To address this challenge, we propose a cross-modal distillation framework that enables effective knowledge transfer across structurally heterogeneous feature spaces via a vector-quantized codebook. Specifically, teacher features are abstracted into a set of vector-form codes regardless of their original feature structure, and the selected codes serve as concept-level anchors for student learning. Code selection is guided by both task relevance and student compatibility, allowing the student to receive transferable teacher knowledge without requiring direct unit-level feature alignment. Experimental results across diverse cross-modal distillation scenarios demonstrate the effectiveness of the proposed framework on classification and semantic segmentation tasks.

1 Introduction

Cross-modal knowledge distillation (CMKD) aims to improve a student model operating on a target modality by transferring knowledge from an auxiliary teacher modality available during training [8]. This setting is practically important when an informative modality can provide rich supervision during training, while the deployed model must rely on a single modality at inference time due to sensing constraints, deployment cost, privacy concerns, or missing data [13, 33].

Unlike single-modality distillation [12], CMKD transfers knowledge between models built on different modalities, whose inputs, representations, and inductive biases can differ. To address this modality gap [18, 34], early approaches [15] focused on selectively distilling task-relevant information that is beneficial to the student modality, primarily at the prediction level. More recent methods [16, 21] have extended this line of work by incorporating not only prediction-level signals but also feature-level cues, such as structural or modality-consistent representations.

While promising, existing approaches inherently assume structural compatibility between teacher and student representations, which enables alignment via projection or shared embedding spaces.

*Corresponding authors.

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

However, such an assumption does not hold in general cross-modal settings, where representation structures can differ fundamentally. For example, visual features are organized as spatial grids, whereas audio and text features are organized as sequences. As illustrated in Fig. 1, the same semantic cue can appear as a spatial pattern in one modality and as a sequential pattern in another. Directly matching feature units across such heterogeneous structures can impose misleading correspondences.

In this paper, we address cross-modal knowledge distillation between intermediate representations with distinct structural organizations. Our key idea is to avoid direct alignment between raw teacher and student feature units, and instead represent teacher-side intermediate knowledge as a discrete set of transferable anchors. Specifically, we learn a vector-quantized teacher codebook over intermediate representations, where the codes serve as anchors that capture recurring task-related patterns in the teacher feature space. Since not all teacher-side codes are equally useful for the student, we estimate the importance of each code by jointly considering its task relevance and its compatibility with the student modality. The selected codes are then used to guide student-side representation learning, allowing the student to exploit fine-grained teacher-side intermediate cues without requiring explicit correspondence between heterogeneous feature units.

We validate the proposed method across diverse cross-modal distillation settings, including audio-visual, image-text, and RGB-depth scenarios. Our experiments cover both classification and segmentation tasks, and compare the proposed framework with prediction-level distillation, feature-level distillation, and modality-gap-aware CMKD baselines. The results demonstrate that the proposed framework can effectively leverage intermediate teacher knowledge even when teacher and student features have structurally different organizations.

Our contributions are summarized as follows:

- We address cross-modal knowledge distillation between intermediate representations with distinct structural organizations, where direct feature alignment is unreliable due to the absence of clear unit-level correspondence between teacher and student features.
- We propose a codebook-guided CMKD framework that learns discrete anchors from teacher intermediate representations, selects transferable teacher-side codes, and uses them to guide student learning without requiring direct feature-unit alignment.
- We validate the proposed framework across diverse cross-modal benchmarks and tasks, including classification and segmentation, demonstrating its effectiveness over prediction-level, feature-level, and modality-gap-aware distillation baselines.

2 Related Work

2.1 Cross-modal Knowledge Distillation

Knowledge distillation (KD) transfers supervision from a teacher model to a student model. A common form of KD encourages the student to mimic the teacher's predictions by matching output distributions or logits [12]. Beyond such prediction-level transfer, many studies have explored richer supervision in the representation space. MLLD [17] improves distillation through multi-level prediction alignment, FitNets [27] transfers intermediate hint features, RKD [25] distills relational structure among samples, CRD [31] uses contrastive learning to preserve representation-level knowledge, and OFA [9] enables distillation across heterogeneous architectures by projecting intermediate features into an aligned logits-like space.

More recently, cross-modal knowledge distillation (CMKD) has attracted increasing attention in settings where an auxiliary modality is available during training but not at test time. Compared with conventional KD, CMKD is more challenging because teacher and student are built on different modalities and thus encode both shared semantics and modality-specific factors. In this setting, recent studies have emphasized that effective transfer should account for the modality gap between teacher and student, rather than indiscriminately enforcing the student to imitate all teacher signals [34, 18]. C2KD explicitly addresses the modality gap through customized distillation strategies for modality imbalance and soft-label misalignment [15]. STXD reduces cross-modal discrepancy through structural and temporal distillation [16], while FDD disentangles feature components in the frequency domain so that modality-consistent information can be transferred more strongly [21].

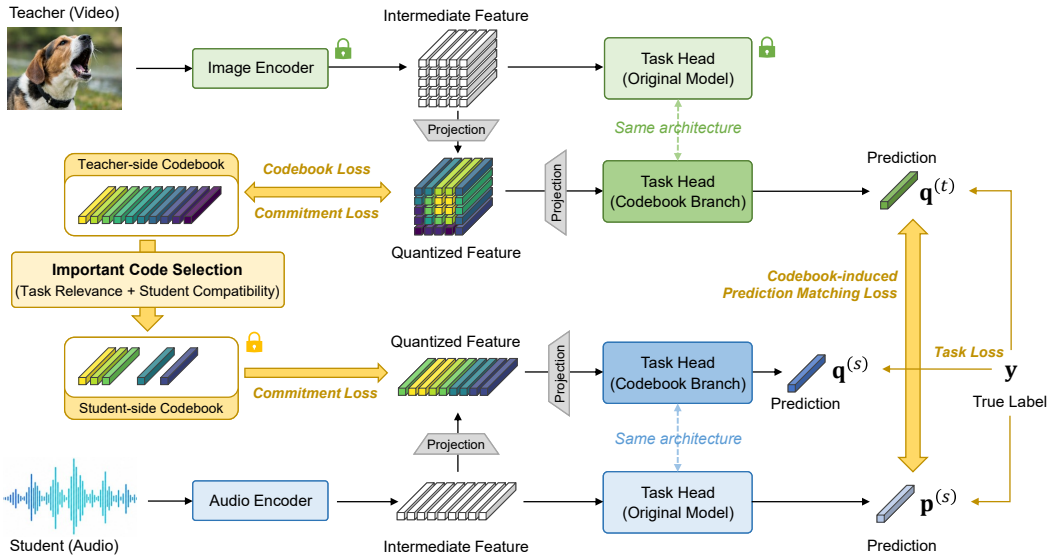


Figure 1: Overview of the proposed framework. We learn a teacher-side vector-quantized codebook (Sec. 3.3), select task-relevant and student-compatible codes (Sec. 3.4), and use the selected codes for student-side distillation (Sec. 3.5). The framework enables feature-level CMKD without requiring explicit unit-level correspondence between structurally different teacher and student representations.

Despite these advances, most CMKD methods implicitly assume that teacher and student representations lie in structurally comparable spaces, enabling alignment via projection or shared embeddings. However, this assumption becomes less straightforward when modalities exhibit substantially different structures. Consequently, cross-modal distillation across structurally heterogeneous representations remains a challenging problem. In this work, we address this problem with a codebook-based distillation framework that abstracts teacher representations into discrete anchors for student learning.

2.2 Codebook Learning and Vector Quantization

Vector quantization (VQ) maps continuous representations to a finite set of learned codewords by assigning each feature to its nearest code. This discrete abstraction was popularized by VQ-VAE, which learns a codebook with codebook and commitment objectives [32], and has since been extended through hierarchical quantization and high-resolution visual representation learning [7, 26]. By replacing continuous features with reusable codewords, VQ provides a compact representation space that can capture recurring patterns in intermediate features.

Beyond compression and reconstruction, recent studies have explored codebooks as semantic interfaces for multimodal representation learning. Language-guided codebook learning aligns visual codes with textual semantics [20], while unified codebook learning has been explored for multimodal large language models [36]. Conceptual codebooks further connect visual representations with conceptual language prompts [35], and recent work on multimodal unified representations refines discrete code spaces for cross-modal generalization [14]. These studies suggest that learned codebooks can organize continuous feature-level information into compact and semantically meaningful discrete units.

However, these studies do not specifically address cross-modal knowledge distillation between structurally heterogeneous intermediate features, where teacher and student feature units lack direct correspondence. In contrast, our work uses a teacher-side VQ codebook as a distillation interface, abstracting teacher representations into discrete concept anchors and selecting transferable codes based on task relevance and student compatibility. This enables feature-level teacher knowledge to guide the student without imposing explicit unit-to-unit alignment across heterogeneous modalities.

3 Proposed Method

3.1 Notations and Problem Statement

We consider a cross-modal knowledge distillation setting with paired teacher-modality (t) and student-modality (s) inputs and corresponding targets,

$$\mathbf{X}^{(t)} = \{\mathbf{x}_i^{(t)}\}_{i=1}^N, \qquad \mathbf{X}^{(s)} = \{\mathbf{x}_i^{(s)}\}_{i=1}^N, \qquad \mathbf{Y} = \{\mathbf{y}_i\}_{i=1}^N, \tag{1}$$

where N denotes the number of training samples. The same subscript i indicates that $\mathbf{x}_i^{(t)}$, $\mathbf{x}_i^{(s)}$, and $\mathbf{y}_i$ correspond to the same sample (i.e., the i-th sample). For each modality $m \in \{t, s\}$, let

$$\mathbf{z}_i^{(m)} = f^{(m)}(\mathbf{x}_i^{(m)}; \theta^{(m)}) \in \mathbb{R}^{L^{(m)} \times D^{(m)}}, \qquad \mathbf{p}_i^{(m)} = g^{(m)}(\mathbf{z}_i^{(m)}; \phi^{(m)}), \tag{2}$$

where $f^{(m)}$ and $g^{(m)}$ denote the modality-specific encoder and task head, respectively. Here, $\mathbf{z}_i^{(m)}$ represents the modality-specific feature representation, and $\mathbf{p}_i^{(m)}$ denotes the corresponding task prediction (e.g., class probabilities in classification). $D^{(m)}$ is the feature dimension and $L^{(m)}$ is the number of modality-specific structured feature units. Because the teacher and student are built on different modalities, their intermediate features may have different structural organizations and sizes. For example, in a 2D visual feature map of image data, $L^{(m)}$ may denote the number of spatial locations (i.e., width $\times$ height), whereas in a 1D sequential feature of audio data, $L^{(m)}$ may denote the number of temporal positions.

Given paired cross-modal training data and a pretrained teacher model $\{\theta^{*(t)}, \phi^{*(t)}\}$, our goal is to learn the student model $\{\theta^{(s)}, \phi^{(s)}\}$ by transferring useful knowledge from the teacher modality to improve task performance of the student model.

3.2 Overall Framework

The proposed CMKD framework for heterogeneously structured features is illustrated in Fig. 1. The proposed framework consists of three stages: teacher-side codebook learning, code-wise importance estimation, and selective codebook-based distillation. First, we introduce a teacher-side codebook branch and learn a vector-quantized codebook that captures task-related patterns in the teacher feature space. (Sec. 3.3) Second, we estimate the importance of each code by jointly considering task relevance and student compatibility. (Sec. 3.4) Third, we select high-importance codes and use them as a codebook for student-side quantization, enabling feature-level distillation without explicit unit-level correspondence between teacher and student representations. We further perform prediction-level alignment between the student and the teacher-side codebook-branch prediction during student training. (Sec. 3.5)

3.3 Stage-1: Teacher-Side Codebook Learning

In this stage, we freeze the pretrained teacher model with parameters $\{\theta^{*(t)}, \phi^{*(t)}\}$. For each teacher-modality input $\mathbf{x}_i^{(t)}$, we extract the intermediate feature $\mathbf{z}_i^{(t)}$ using the fixed teacher encoder:

$$\mathbf{z}_i^{(t)} = f^{(t)}(\mathbf{x}_i^{(t)}; \theta^{*(t)}). \tag{3}$$

To obtain a discrete abstraction of $\mathbf{z}_i^{(t)}$, we introduce a learnable codebook $\mathbf{C} \in \mathbb{R}^{B \times D}$ with two projection layers, $\Pi_{\text{in}}^{(t)}$ and $\Pi_{\text{out}}^{(t)}$, which project the feature into the code space and reconstruct it back to the original feature space, respectively. The reconstructed feature is then fed into a teacher-side codebook-branch prediction head $g_{\text{cp}}^{(t)}(\cdot; \phi_{\text{cp}}^{(t)})$, which follows the architecture of the original teacher task head but has separate learnable parameters. The overall forward process is given by

$$\mathbf{e}_i^{(t)} = \text{VQ}\Big(\Pi_{\text{in}}^{(t)}(\mathbf{z}_i^{(t)}); \mathbf{C}\Big), \qquad \mathbf{q}_i^{(t)} = g_{\text{cp}}^{(t)}\Big(\Pi_{\text{out}}^{(t)}(\mathbf{e}_i^{(t)}); \phi_{\text{cp}}^{(t)}\Big), \tag{4}$$

where vector quantization (VQ) assigns each projected token to its nearest codeword in $\mathbf{C}$, thereby converting the continuous teacher feature into a discrete code-based representation [32]. Here, $\mathbf{e}_i^{(t)}$ denotes the quantized representation, and $\mathbf{q}_i^{(t)}$ denotes the prediction from the codebook branch.

To train the teacher-side codebook branch, we optimize the task loss together with the standard vector-quantization objective, consisting of the codebook loss and the commitment loss [32]:

$$\mathcal{L}_{\mathrm{S1}}^{\mathrm{code}} = \sum_{i=1}^{N} \left(\mathcal{L}_{\mathrm{task}}(\mathbf{q}_i^{(t)}, \mathbf{y}_i) + \left\| \mathrm{sg}\left[\Pi_{\mathrm{in}}^{(t)}(\mathbf{z}_i^{(t)}) \right] - \mathbf{e}_i^{(t)} \right\|_2^2 + \beta_{\mathrm{com}} \left\| \Pi_{\mathrm{in}}^{(t)}(\mathbf{z}_i^{(t)}) - \mathrm{sg}\left[\mathbf{e}_i^{(t)} \right] \right\|_2^2 \right), \quad (5)$$

where $\mathrm{sg}[\cdot]$ denotes the stop-gradient operator, β_{com} is the commitment loss weight and $\mathcal{L}_{\mathrm{task}}$ denotes the task-specific loss (e.g., cross-entropy for classification).

For effective student distillation, the learned teacher codebook should preserve task-relevant information while remaining compatible with the student modality. We therefore jointly train the teacher-side codebook branch together with the student model. Specifically, let $\mathbf{p}_i^{(s)}$ denote the prediction from the original student task head. We introduce a *codebook-induced prediction matching* term that aligns $\mathbf{q}_i^{(t)}$ with $\mathbf{p}_i^{(s)}$. The overall objective is given by

$$\mathcal{L}_{\mathrm{S1}} = \mathcal{L}_{\mathrm{S1}}^{\mathrm{code}} + \sum_{i=1}^{N} \left(\mathcal{L}_{\mathrm{task}}(\mathbf{p}_i^{(s)}, \mathbf{y}_i) + \left\| \mathbf{q}_i^{(t)} - \mathbf{p}_i^{(s)} \right\|_2^2 \right). \quad (6)$$

3.4 Stage-2: Code-wise Importance Estimation

Although Eq. 6 encourages the teacher-side codebook to learn codes that are compatible with the student modality, not all codes are equally beneficial for transfer. We therefore estimate the importance of each code and select a subset of high-importance codes for subsequent distillation.

To estimate code importance, we use the teacher-side codebook branch learned in Stage-1 and evaluate how its prediction changes when each code is removed. For each code k, we form a *leave-one-code-out* codebook $\mathbf{C}_{-k}$ by removing it from $\mathbf{C}$ and compute the corresponding teacher-side codebook-branch prediction:

$$\bar{\mathbf{e}}_i^{(t)} = \mathrm{VQ}\left(\Pi_{\mathrm{in}}^{(t)}(\mathbf{z}_i^{(t)}); \mathbf{C}_{-k} \right), \qquad \bar{\mathbf{q}}_i^{(t)} = g_{\mathrm{cp}}^{(t)}\left(\Pi_{\mathrm{out}}^{(t)}(\bar{\mathbf{e}}_i^{(t)}); \phi_{\mathrm{cp}}^{(t)} \right). \quad (7)$$

We then estimate the importance of the excluded code from two perspectives: task relevance and student compatibility,

$$u_k^{\mathrm{rel}} = \mathbb{E}_i \left[\mathcal{L}_{\mathrm{task}}(\bar{\mathbf{q}}_i^{(t)}, \mathbf{y}_i) - \mathcal{L}_{\mathrm{task}}(\mathbf{q}_i^{(t)}, \mathbf{y}_i) \right], u_k^{\mathrm{comp}} = \mathbb{E}_i \left[\left\| \bar{\mathbf{q}}_i^{(t)} - \mathbf{p}_i^{(s)} \right\|_2^2 - \left\| \mathbf{q}_i^{(t)} - \mathbf{p}_i^{(s)} \right\|_2^2 \right], \quad (8)$$

where $\mathbf{q}_i^{(t)}$ is the teacher-side codebook branch prediction obtained with the full codebook $\mathbf{C}$. The relevance score u_k^{rel} measures how much removing the k-th code degrades the teacher-side task prediction. A larger value indicates that the code is more important for preserving task-relevant information. The compatibility score u_k^{comp} measures how much removing the code increases the discrepancy between the teacher-side codebook-branch prediction and the student prediction. A larger value indicates that the code provides guidance that is more compatible with the student modality. After normalizing the two scores, we compute the final code importance by

$$u_k = u_k^{\mathrm{rel}} + u_k^{\mathrm{comp}}. \quad (9)$$

The resulting scores are used to rank teacher codes and select a subset for the next distillation stage.

3.5 Stage-3: Selective Codebook-Based Distillation

In the final stage, we select the top-K codes according to the importance scores obtained in Stage-2 and construct the selected teacher codebooks $\mathbf{C}_{\mathrm{sel}}$, which is used as teacher-derived anchors for student-side quantization. Specifically, we build a student codebook branch with learnable projection layers $\Pi_{\mathrm{in}}^{(s)}$ and $\Pi_{\mathrm{out}}^{(s)}$, along with a student-side codebook-branch prediction head $g_{\mathrm{cp}}^{(s)}(\cdot; \phi_{\mathrm{cp}}^{(s)})$. Through this branch, student features are quantized using the selected teacher codebook, allowing teacher-side intermediate knowledge to guide student learning without direct feature-unit alignment.

Given the student intermediate feature $\mathbf{z}_i^{(s)}$, the student-side codebook-branch produces

$$\mathbf{e}_i^{(s)} = \mathrm{VQ}\left(\Pi_{\mathrm{in}}^{(s)}(\mathbf{z}_i^{(s)}); \mathbf{C}_{\mathrm{sel}} \right), \qquad \mathbf{q}_i^{(s)} = g_{\mathrm{cp}}^{(s)}\left(\Pi_{\mathrm{out}}^{(s)}(\mathbf{e}_i^{(s)}); \phi_{\mathrm{cp}}^{(s)} \right), \quad (10)$$

Table 1: Classification results. A, V, T, and I denote audio, video, text, and image modalities, respectively, and each arrow indicates the teacher-to-student distillation direction. We report accuracy averaged over five independent runs, with the best result in each column shown in bold.

Method	RAVDESS		VGG-Sound		CrisisMMD		MM-IMDB	
	A → V	V → A	A → V	V → A	T → I	I → T	T → I	I → T
w/o KD	0.9208	0.6826	0.5031	0.6058	0.5403	0.5500	0.6645	0.7294
General distillation baselines								
KLD	0.9118	0.6972	0.5153	0.6176	0.5377	0.5578	0.6732	0.7310
MLLD	0.9292	0.6889	0.5135	0.6122	0.5357	0.5562	0.6699	0.7320
FitNets	0.9062	0.7146	0.4916	0.5962	0.5281	0.5507	0.6759	0.7296
RKD	0.9236	0.7014	0.5066	0.6085	0.5432	0.5569	0.6668	0.7317
CRD	0.9299	0.7007	0.5063	0.6110	0.5377	0.5560	0.6685	0.7299
OFA	0.9215	0.6562	0.5100	0.6129	0.5362	0.5565	0.6715	0.7339
Modality-gap-aware CMKD baselines								
MGDFR	0.9313	0.7264	0.5225	0.6186	**0.5440**	0.5584	0.6725	0.7382
C2KD	0.9306	0.7021	0.5123	0.6154	0.5410	0.5520	0.6670	0.7372
STXD	0.9035	0.7250	0.5067	0.6093	0.5421	0.5485	0.6761	0.7265
FDD	0.9153	0.7055	0.5028	0.6063	0.5382	0.5547	0.6670	0.7323
Ours	**0.9431**	**0.7486**	**0.5299**	**0.6229**	0.5433	**0.5592**	**0.6768**	**0.7434**

where $\mathbf{C}_{\text{sel}}$ is fixed during this stage. Since the codebook is no longer updated, the codebook branch is trained only with the task loss and the commitment loss:

$$\mathcal{L}_{\text{S3}}^{\text{code}} = \mathcal{L}_{\text{task}}(\mathbf{q}_i^{(s)}, \mathbf{y}_i) + \beta_{\text{com}} \left\| \Pi_{\text{in}}^{(s)}(\mathbf{z}_i^{(s)}) - \text{sg}\left[\mathbf{e}_i^{(s)}\right] \right\|_2^2. \tag{11}$$

The teacher-side codebook branch learned in Stage-1 produces a codebook-induced prediction that reflects quantized teacher knowledge. To fully leverage the learned teacher codebook, we use its induced prediction to guide the student task prediction, allowing the codebook to jointly shape student-side quantization and final prediction learning. Specifically, we introduce a *codebook-induced prediction matching* term between $\mathbf{q}_i^{(t)}$ and $\mathbf{p}_i^{(s)}$, leading to the Stage-3 objective:

$$\mathcal{L}_{\text{S3}} = \mathcal{L}_{\text{task}}(\mathbf{p}_i^{(s)}, \mathbf{y}_i) + \lambda_{\text{dist}} \left(\mathcal{L}_{\text{S3}}^{\text{code}} + \left\| \mathbf{q}_i^{(t)} - \mathbf{p}_i^{(s)} \right\|_2^2 \right), \tag{12}$$

where λ_{dist} controls the contribution of the distillation terms. Through this stage, the student is trained with the original task objective and two codebook-based distillation signals, which guide both student intermediate representations and task predictions using the learned teacher codebook, without requiring direct alignment between structurally heterogeneous features. Algorithm 1 in Appendix C provides the pseudo-code of the proposed method.

4 Experiments

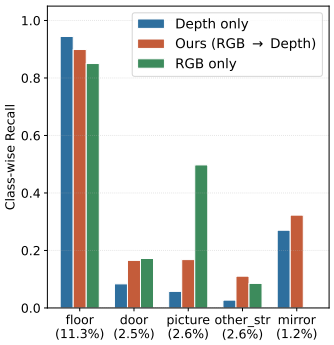
We validate the proposed method on five multimodal benchmarks spanning classification and semantic segmentation tasks in CMKD settings. Implementation details are provided in the Appendix A.

4.1 Classification Experiments

Datasets. We use four multimodal classification benchmarks. RAVDESS [22] contains paired speech audio and facial video recordings for eight-class emotion recognition. VGG-Sound [4] is a video–audio event classification dataset, where we use the curated 141-class subset following [19]. CrisisMMD-v2.0 [1, 24] consists of disaster-related image–text social media posts for eight-class humanitarian category classification. MM-IMDB [2] contains paired movie posters and plot descriptions for movie genre classification. For all datasets, we use a 6:2:2 train/validation/test split and report test accuracy from the best validation checkpoint, averaged over five runs.

Table 2: Semantic segmentation results on NYU-Depth. **Left**: quantitative comparison for two distillation scenarios, where the first modality denotes the teacher and the second denotes the student. **Right**: class-wise recall comparison of RGB-only, depth-only, and our RGB → Depth student models.

Method	Depth → RGB			RGB → Depth		
	OA ↑	CA ↑	mIoU ↑	OA ↑	CA ↑	mIoU ↑
w/o KD	0.5582	0.2748	0.1850	**0.5269**	0.2100	0.1366
MLLD	0.5599	0.2792	0.1911	0.5250	0.2118	0.1399
CRD	0.5592	0.2677	0.1806	0.5261	0.2111	0.1385
OFA	0.5607	0.2881	0.1982	0.5242	0.2123	0.1409
MGDFR	0.5601	0.2737	0.1878	0.5213	0.2162	0.1442
C2KD	0.5587	0.2771	0.1878	0.5214	0.2093	0.1396
STXD	0.5616	0.2786	0.1925	0.5231	0.2164	0.1425
FDD	0.5608	0.2744	0.1855	0.5252	0.2122	0.1403
Ours	**0.5632**	**0.2909**	**0.2003**	0.5218	**0.2184**	**0.1456**



Baselines. We compare our method with general distillation methods including KLD [12] and MLLD [17], FitNets [27], RKD [25], CRD [31], and OFA [9]; and modality-gap-aware CMKD methods, including MGDFR [34], C2KD [15], STXD [16], and FDD [21]. For feature-alignment baselines, which are not directly applicable to structurally heterogeneous features, we use pooled representations and add projection layers when feature dimensions differ.

Implementation. Image and video inputs are represented as 2D spatial feature using ResNet- and ViT-based feature extractors [6, 10]. Audio inputs are represented as 1D temporal feature sequences using Wav2Vec2- and VGGish-style feature extractors [3, 11]. Text inputs are represented as token-level sequence features using transformer-based feature extractors including BERTweet and FLAVA [23, 30]. We select the top 50% of teacher codes according to the proposed importance score.

Results. Tab. 1 summarizes the classification results. Our method achieves the best performance in seven out of eight settings and the second-best result on CrisisMMD T → I. It also improves over the student trained without distillation in all transfer directions, showing that the proposed framework provides useful supervision across both audio-visual and text-image scenarios. Modality-gap-aware CMKD baselines are generally competitive, which supports the importance of accounting for cross-modal discrepancy. Nevertheless, our method obtains stronger results in most settings, suggesting that useful teacher-side cues can be extracted from structurally heterogeneous intermediate features and transferred through a codebook-level interface.

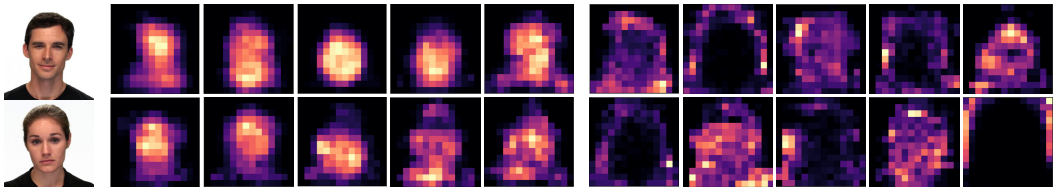
4.2 Semantic Segmentation Experiments

Dataset. We evaluate semantic segmentation on NYU-Depth V2 [28], an indoor RGB-D benchmark with aligned RGB-depth image pairs and pixel-wise annotations for 40 classes. Following [19], we report Overall Accuracy (OA), Class-averaged Accuracy (CA), and mean Intersection-over-Union (mIoU), which measure pixel-level accuracy, average class-wise recall, and average region overlap, respectively. We use a 6:2:2 train/validation/test split and report results averaged over five runs.

Baselines. We use representative baselines from the classification experiments. Since RGB and depth inputs are spatially aligned, dense correspondence between teacher and student feature maps is available. Thus, feature-level baselines are applied directly to corresponding intermediate feature maps when applicable, with projection layers added only to match channel dimensions.

Implementation. We use a ResNet-34 [10] based feature extractor for both RGB and depth modalities, and apply our method to the resulting 14×14 grid features. The codebook contains 1024 codes, from which the top 12.5% are selected based on the estimated importance scores.

Results. Tab. 2 reports the semantic segmentation results. The left part provides quantitative comparisons, and the right part shows class-wise recall for selected categories in the RGB→Depth setting. Our method achieves the best performance on most metrics, with the only exception being OA in



(a) Example (b) Spatial usage maps of top-ranked codes (c) Spatial usage maps of bottom-ranked codes

Figure 2: Codebook analysis on RAVDESS. (a) Example frames from the dataset. (b,c) Spatial usage maps of top-ranked and bottom-ranked codes according to the proposed importance score.

RGB→Depth, where the w/o KD baseline remains slightly higher. Since OA is weighted by pixel frequency, recall drops in high-occupancy categories can noticeably affect the overall score (Appendix B.5). The decrease on `floor` in the class-wise recall comparison provides one representative example. Meanwhile, our method improves CA and mIoU, with recall gains on categories such as `door`, `picture`, and `other_structure`, where depth-only features may provide less discriminative cues but can benefit from guidance provided by the RGB teacher. These results suggest that the proposed method can transfer useful teacher-side information for class-level segmentation quality.

Table 3: Ablation study on Stage-3 loss components. "Code" denotes the student-side codebook-branch objective. "Pred-Match" denotes the codebook-induced prediction matching term.

Code	Pred Match	RAVDESS		VGG-Sound		CrisisMMD		MM-IMDB	
		A → V	V → A	A → V	V → A	T → I	I → T	T → I	I → T
-	-	0.9208	0.6826	0.5031	0.6058	0.5403	0.5500	0.6645	0.7294
✓	-	0.9202	0.7333	0.5067	0.6098	0.5387	0.5534	0.6716	0.7363
-	✓	0.9285	0.7479	0.5294	0.6204	0.5367	0.5529	0.6770	0.7421
✓	✓	0.9431	0.7486	0.5299	0.6229	0.5433	0.5592	0.6768	0.7434

Table 4: Ablation on the prediction matching target in Stage-3 Loss.

Pred-Match Target	RAVDESS		VGG-Sound		CrisisMMD		MM-IMDB	
	A → V	V → A	A → V	V → A	T → I	I → T	T → I	I → T
Original branch ($\mathbf{p}^{(t)}$)	0.8903	0.6201	0.5173	0.6171	0.5425	0.5539	0.6654	0.7351
Codebook branch ($\mathbf{q}^{(t)}$)	0.9431	0.7486	0.5299	0.6229	0.5433	0.5592	0.6768	0.7434

4.3 Analysis

Do High-Importance Codes Capture Informative Patterns? We further analyze the learned codebook on RAVDESS, where the actor faces are well aligned across videos, making the spatial usage patterns of codes easier to interpret. We consider the visual-teacher setting and visualize where each code is assigned over the spatial feature grid across the dataset. Fig. 2 shows the usage maps of the top-10 and bottom-10 codes according to the importance score. Since RAVDESS requires emotion recognition from acted facial and vocal expressions, task-relevant visual concepts are expected to be concentrated around centered regions, especially the face and mouth. Consistent with this intuition, high-score codes are mainly activated around the central regions, suggesting that they capture visual concepts that are informative to the task and likely useful for cross-modal transfer. In contrast, low-score codes tend to show more scattered or peripheral usage patterns, indicating that they are less consistently associated with meaningful regions.

Effect of Distillation Loss Components. Tab. 3 analyzes the contribution of the two main Stage-3 loss components in Eq. 12. "Code" denotes the student-side codebook-branch objective in Eq. (11), which constrains student features through the selected teacher codebook and provides representation-level guidance. "Pred-Match" denotes the codebook-induced prediction matching term in Eq. (12), which guides the student task prediction using the teacher prediction derived from quantized representations. While each component improves several transfer directions, combining them yields the

Table 5: Ablation study on code selection criteria.

Selection Criteria	RAVDESS		VGG-Sound		CrisisMMD		MM-IMDB	
	A → V	V → A	A → V	V → A	T → I	I → T	T → I	I → T
Random	0.9292	0.7396	0.5252	0.6211	0.5353	0.5536	0.6718	0.7353
Task relevance only	0.9396	0.7361	0.5276	0.6233	0.5435	0.5558	0.6727	0.7351
Student compatibility only	0.9375	0.7417	0.5275	0.6222	0.5437	0.5565	0.6722	0.7346
Combination (Ours)	0.9431	0.7486	0.5299	0.6229	0.5433	0.5592	0.6768	0.7434

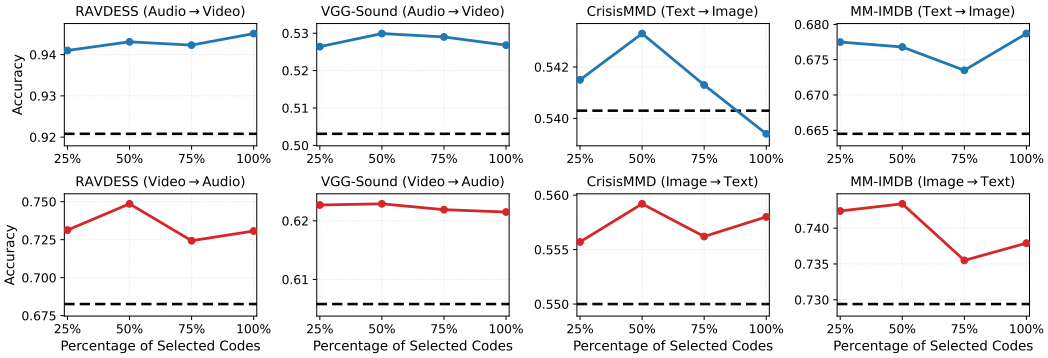


Figure 3: Selected code ratio ablation. We vary the percentage of selected teacher codes used for student-side quantization from 25% to 100%. The dashed line denotes the w/o KD baseline.

strongest overall performance, suggesting complementary effects between representation guidance and codebook-induced prediction guidance.

Tab. 4 further examines the target used for prediction matching by comparing the original teacher prediction $\mathbf{p}^{(t)}$ with the codebook-induced teacher prediction $\mathbf{q}^{(t)}$. Matching $\mathbf{q}^{(t)}$ outperforms matching $\mathbf{p}^{(t)}$ across all evaluated datasets and transfer directions, indicating that the gain from "Pred-Match" does not come from generic output-level supervision alone, but from using a guidance signal shaped by the learned teacher codebook.

Effect of Code Selection. Fig. 3 analyzes the effect of the selected code ratio by varying the percentage of teacher codes used for student-side quantization from 25% to 100%, where the dashed line denotes the w/o KD baseline. The proposed method remains above the w/o KD baseline in most cases, indicating that its benefit is not limited to a single carefully tuned ratio. The best ratio varies across settings, suggesting that the amount of useful teacher-side code information can depend on the dataset and transfer direction. In the main classification experiments in Sec. 4.1, we use a fixed ratio of 50% for all settings to maintain a simple and consistent evaluation protocol (Appendix B.4).

Tab. 5 further analyzes the effect of different code selection criteria under the fixed 50% selection ratio used in Sec. 4.1. Compared with random selection, score-based criteria generally improve performance, indicating that the learned scores provide useful signals for selecting teacher codes. When codes are selected based only on task relevance or only on student compatibility score, the two criteria show different strengths across scenarios. Using their combined score achieves the best or competitive performance in most cases, suggesting that jointly considering task relevance and student compatibility provides a more balanced criterion for identifying useful teacher codes.

5 Conclusion

We presented a codebook-guided framework for cross-modal knowledge distillation between structurally heterogeneous features. Instead of directly aligning teacher and student feature units, our method abstracts teacher representations into a vector-quantized codebook and selects task-relevant and student-compatible codes for student learning. The selected codes are then fully leveraged through student-side codebook quantization and codebook-induced prediction matching. This design allows the student to exploit fine-grained teacher-side information without requiring explicit unit-level correspondence between heterogeneous feature structures, such as spatial grids and temporal sequences. Across diverse cross-modal classification and semantic segmentation settings, our method achieves

strong performance compared with general and modality-gap-aware distillation baselines. Further analyses support the proposed framework design and suggest that codebook-guided distillation is a practical strategy for transferring knowledge across structurally heterogeneous features.

Future work. In the current framework, code selection relies on a fixed ratio and an offline importance estimation stage. The fixed ratio may not capture the varying amount of useful teacher-side information across cross-modal scenarios, while offline scoring can increase the cost for larger codebooks (Appendix B.3). Future work will explore adaptive and efficient strategies for selecting appropriate code subsets according to the distillation scenario.

Acknowledgments

This research was supported by Kyungpook National University Research Fund, 2025. This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2026-25495369). This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) [RS-2021-II211341, Artificial Intelligence Graduate School Program (Chung-Ang University), and RS-2022-II220124, Development of Artificial Intelligence Technology for Self-Improving Competency-Aware Learning Capabilities]. This research was also supported by the AI Seoul Tech Research Support Program of the Seoul Future Foundation.

Competing Interests

The authors declare no competing financial interests related to this work.

References

[1] Firoj Alam, Ferda Ofli, and Muhammad Imran. Crisismmd: Multimodal twitter datasets from natural disasters. In *Proceedings of the international AAAI conference on web and social media*, volume 12, 2018.

[2] John Arevalo, Thamar Solorio, Manuel Montes-y Gómez, and Fabio A. González. Gated multimodal units for information fusion. In *Proceedings of the International Conference on Learning Representations Workshop*, 2017.

[3] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460, 2020.

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

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[7] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021.

[8] Saurabh Gupta, Judy Hoffman, and Jitendra Malik. Cross modal distillation for supervision transfer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2827–2836, 2016.

[9] Zhiwei Hao, Jianyuan Guo, Kai Han, Yehui Tang, Han Hu, Yunhe Wang, and Chang Xu. One-for-all: Bridge the gap between heterogeneous architectures in knowledge distillation. *Advances in Neural Information Processing Systems*, 36:79570–79582, 2023.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[11] Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore, Manoj Plakal, Devin Platt, Rif A Saurous, Bryan Seybold, et al. Cnn architectures for large-scale audio classification. In *2017 ieee international conference on acoustics, speech and signal processing (icassp)*, pages 131–135. IEEE, 2017.

[12] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[13] Judy Hoffman, Saurabh Gupta, and Trevor Darrell. Learning with side information through modality hallucination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 826–834, 2016.

[14] Hai Huang, Yan Xia, Shengpeng Ji, Shulei Wang, Hanting Wang, Minghui Fang, Jieming Zhu, Zhenhua Dong, Sashuai Zhou, and Zhou Zhao. Enhancing multimodal unified representations for cross modal generalization. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 2353–2366, 2025.

[15] Fushuo Huo, Wenchao Xu, Jingcai Guo, Haozhao Wang, and Song Guo. C2kd: Bridging the modality gap for cross-modal knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16006–16015, 2024.

[16] Sujin Jang, Dae Ung Jo, Sung Ju Hwang, Dongwook Lee, and Daehyun Ji. Stxd: Structural and temporal cross-modal distillation for multi-view 3d object detection. *Advances in Neural Information Processing Systems*, 36:29323–29342, 2023.

[17] Ying Jin, Jiaqi Wang, and Dahua Lin. Multi-level logit distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24276–24285, 2023.

[18] Dae Ung Jo, Sujin Jang, Jay Heo, and Sung Ju Hwang. Understanding dimensional collapse in cross-modal feature distillation. *OpenReview*, 2025. ICLR 2025 submission.

[19] Hui Li, Pengfei Yang, Juanyang Chen, Le Dong, Yanxin Chen, and Quan Wang. Mst-distill: Mixture of specialized teachers for cross-modal knowledge distillation. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 1588–1597, 2025.

[20] Guotao Liang, Baoquan Zhang, Yaowei Wang, Xutao Li, Yunming Ye, Huaibin Wang, Chuyao Luo, Kola Ye, et al. Lg-vq: Language-guided codebook learning. *arXiv preprint arXiv:2405.14206*, 2024.

[21] Junhong Liu, Yuan Zhang, Tao Huang, Wenchao Xu, and Renyu Yang. Distilling cross-modal knowledge via feature disentanglement. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 23739–23747, 2026.

[22] Steven R Livingstone and Frank A Russo. The ryerson audio-visual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in north american english. *PloS one*, 13(5):e0196391, 2018.

[23] Dat Quoc Nguyen, Thanh Vu, and Anh-Tuan Nguyen. Bertweet: A pre-trained language model for english tweets. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 9–14, 2020.

[24] Ferda Ofli, Firoj Alam, and Muhammad Imran. Analysis of social media data using multimodal deep learning for disaster response. *arXiv preprint arXiv:2004.11838*, 2020.

[25] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3967–3976, 2019.

[26] Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32, 2019.

[27] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2015.

[28] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European conference on computer vision*, pages 746–760. Springer, 2012.

[29] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[30] Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. Flava: A foundational language and vision alignment model. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15638–15650, 2022.

[31] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. *arXiv preprint arXiv:1910.10699*, 2019.

[32] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

[33] Vladimir Vapnik and Akshay Vashist. A new learning paradigm: Learning using privileged information. *Neural networks*, 22(5-6):544–557, 2009.

[34] Zihui Xue, Zhengqi Gao, Sucheng Ren, and Hang Zhao. The modality focusing hypothesis: Towards understanding cross-modal knowledge distillation. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023.

[35] Yi Zhang, Ke Yu, Siqi Wu, and Zhihai He. Conceptual codebook learning for vision-language models. In *European Conference on Computer Vision*, pages 235–251. Springer, 2024.

[36] Sipeng Zheng, Bohan Zhou, Yicheng Feng, Ye Wang, and Zongqing Lu. Unicode: Learning a unified codebook for multimodal large language models. In *European Conference on Computer Vision*, pages 426–443. Springer, 2024.

[37] Jinxing Zhou, Dan Guo, and Meng Wang. Contrastive positive sample propagation along the audio-visual event line. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(6):7239–7257, 2022.

A Implementation Details

A.1 Computational Resources

All experiments were conducted on a single workstation equipped with an AMD Ryzen 9 9900X CPU, an NVIDIA RTX PRO 6000 Blackwell Workstation Edition GPU with 96GB memory, and 64GB RAM. Unless otherwise specified, each training run was performed on a single GPU.

Table 6: Implementation details for each dataset.

Dataset	Epochs	Learning rate	Weight decay	Batch size	#Code	Code Dim.	β_{com}	λ_{dist}
RAVDESS	100	1×10^{-3}	1×10^{-4}	64	512	128	0.25	1×10^{-2}
VGG-Sound	100	1×10^{-3}	1×10^{-4}	128	512	128	0.25	1×10^{-2}
CrisisMMD	100	1×10^{-3}	1×10^{-4}	64	256	128	0.25	1×10^{-2}
MM-IMDB	100	1×10^{-3}	1×10^{-4}	64	512	64	0.25	1×10^{-2}
NYU-Depth V2	100	1×10^{-4}	1×10^{-4}	6	1024	256	0.25	1×10^{-2}

A.2 General Training Details

Table 6 summarizes the training settings used in our experiments. Unless otherwise specified, we train each model using AdamW with $\beta = (0.9, 0.999)$. For a fair comparison, additional training stages required by FitNets [27], MGDFR [34], and our method are also trained for 100 epochs. These correspond to the hint regression stage of FitNets, the co-training stage of MGDFR, and Stage-1 of our method, respectively.

A.3 Implementation Details for Distillation

For our method, codebook sizes (#Code), dimension of each code (Code Dim.), commitment coefficient β_{com} and distillation loss coefficient λ_{dist} are summarized in Table 6. We use a linear projector to map the intermediate features into the code space before vector quantization. Quantization is performed by assigning each projected feature to its nearest code according to the squared Euclidean distance. During Stage-1, we additionally include a small entropy-based code usage regularization term to stabilize codebook learning. Before Stage-3, we re-initialize the student parameters and train the final student from scratch using the selected teacher codebook.

For feature-alignment baselines, we use pooled features for feature-level distillation when the teacher and student intermediate features have different structural forms, and add a projection layer if necessary. When dense correspondence between teacher and student features is available, as in NYU-Depth V2, we apply feature-level distillation directly to the corresponding intermediate features without pooling.

A.4 RAVDESS

Dataset Specification. We use the Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) for audio-visual emotion classification [22]. In our experiments, we use the speech subset and construct paired speech-audio and speech-video samples. The task is formulated as an 8-way emotion classification problem with the following emotion categories: *neutral*, *calm*, *happy*, *sad*, *angry*, *fearful*, *disgust*, and *surprised*. Each paired audio-video sample shares the same emotion label. The resulting dataset contains 1,440 paired samples, which are split into training, validation, and test sets with a ratio of 60%/20%/20%.

Preprocessing. For audio data, we convert each waveform to mono when necessary, resample it to 16 kHz, and normalize its duration to 3.6 seconds. Longer clips are center-cropped, while shorter clips are symmetrically zero-padded. The resulting waveform is passed through a pretrained Wav2Vec2-Base model [3], and the final hidden representations are used as the audio feature sequence. The extracted audio features are arranged as $D \times T = 768 \times 179$, where D denotes the feature dimension and T denotes the temporal length. For video data, we uniformly sample 10 frames from each video. Each frame is center-cropped, resized to 224×224, and normalized using ImageNet statistics [5]. We then feed each frame into an ImageNet-pretrained ResNet-18 [10] and use the layer-3 feature

map as the frame-level visual representation. The frame-level features are averaged over the sampled frames to obtain a single spatial feature map for each video. The extracted visual features have size $D \times H \times W = 256 \times 14 \times 14$, where D denotes the channel (feature) dimension and $H \times W$ denotes the spatial resolution.

Model. We use lightweight classifiers on top of the pre-extracted audio and video features. The audio classifier receives Wav2Vec2 features and applies two 1D convolutional blocks. The resulting encoder output, with shape $B \times 256 \times T$, is used as the audio intermediate feature, where B denotes batch size. It is average-pooled over the temporal dimension and passed through a linear layer with ReLU activation and an 8-way classifier. The video classifier receives ResNet-18 layer-3 features and applies two 2D convolutional blocks. The resulting encoder output, with shape $B \times 256 \times H \times W$, is used as the video intermediate feature, where B denotes batch size. It is average-pooled over the spatial dimensions and passed through a linear layer with ReLU activation and an 8-way classifier. In both classifiers, each convolutional block consists of convolution, batch normalization, and ReLU activation, and the hidden dimension before the final classifier is set to 64.

A.5 VGG-Sound

Dataset Specification. VGG-Sound [4] is a large-scale audio-visual event classification dataset consisting of short audio-video clips collected from YouTube videos. Following [37, 19], we use a subset of 48,755 paired samples covering 141 real-world event classes. Each paired audio-video sample shares the same event label, and the task is formulated as 141-way event classification. We split the dataset into training, validation, and test sets with a ratio of 60%/20%/20%.

Preprocessing. Following the preprocessing protocol of [37], the visual features are extracted using VGG19 [29] and the audio features are extracted using VGGish [11]. Each audio feature is represented as a temporal sequence of size $T \times D = 10 \times 128$, where T denotes the temporal length and D denotes the feature dimension. For visual features, the original pre-extracted representation contains frame-level spatial features. We average the visual features over the temporal dimension and use the resulting spatial feature map of size $H \times W \times D = 7 \times 7 \times 512$, where $H \times W$ denotes the spatial resolution and D denotes the channel dimension.

Model. We use lightweight modality-specific classifiers on top of the pre-extracted features. The audio classifier applies two 1D convolutional blocks to obtain the audio intermediate feature of shape $B \times 128 \times T$, where B denotes batch size. This intermediate feature is further processed by an additional 1D convolutional stage, average-pooled over time, and passed through a linear layer with ReLU activation and a 141-way classifier. The video classifier applies a 1×1 convolutional reduction layer to obtain the video intermediate feature of shape $B \times 128 \times H \times W$. The intermediate feature is further processed by a 2D convolutional block, average-pooled over the spatial dimensions, and passed through a linear layer with ReLU activation and a classifier. In both classifiers, convolutional blocks consist of convolution, batch normalization, and ReLU activation. We set the hidden dimension before the final classifier to 128.

A.6 CrisisMMD

Dataset Specification. We use CrisisMMD-V2 [24], an updated version of the CrisisMMD multimodal disaster dataset [1], for image-text tweet classification in humanitarian crisis settings. The dataset contains 16,058 paired image-text samples, and the task is formulated as 8-way humanitarian category classification. The categories are *not_humanitarian*, *affected_individuals*, *infrastructure_and_utility_damage*, *injured_or_dead_people*, *missing_or_found_people*, *rescue_volunteering_or_donation_effort*, *vehicle_damage*, and *other_relevant_information*. Each paired image-text sample shares the same label. We split the dataset into training, validation, and test sets with a ratio of 60%/20%/20%.

Preprocessing. For image data, we resize each image to 224×224 and normalize it using ImageNet statistics [5]. The processed image is passed through an ImageNet-pretrained ResNet-34 [10], and we use the layer-3 feature map as the image representation. The extracted image features have size $D \times H \times W = 256 \times 14 \times 14$, where D denotes the channel dimension and $H \times W$ denotes the spatial resolution. For text data, we use BERTweet [23] to extract token-level representations from the tweet

Table 7: Number of samples per class in the single-label MM-IMDB dataset.

Class	# Samples
Drama	2,681
Comedy	1,337
Documentary	918
Horror	417
Western	173
Thriller	143
Sci-Fi	56
Action	53
Total	5,778

text. Each tweet is tokenized with a maximum length of 128, and the last hidden states of BERTweet are used as the text feature sequence. The extracted text features have size $T_i \times D = T_i \times 768$, where T_i denotes the valid token length of the i-th tweet and D denotes the feature dimension.

Model. We use lightweight modality-specific classifiers on top of the pre-extracted image and text features. The image classifier applies two 2D convolutional blocks. The resulting encoder output, with shape $B \times 128 \times H \times W$, is used as the image intermediate feature, where B denotes batch size. It is average-pooled over the spatial dimensions and passed through a linear layer with ReLU activation and an 8-way classifier. The text classifier applies two 1D convolutional blocks. The resulting encoder output, with shape $B \times 128 \times T$, is used as the text intermediate feature, where T denotes the token sequence length. For text pooling, we average only over valid token positions using the token lengths. The pooled representation is then passed through a linear layer with ReLU activation and an 8-way classifier. In both classifiers, each convolutional block consists of convolution, batch normalization, and ReLU activation, with hidden dimensions set to 64.

A.7 MM-IMDB

Dataset Specification. We use MM-IMDB [2] for image-text movie genre classification. Although the original dataset is a 23-class multi-label genre classification benchmark, we construct a single-label subset for our experiments. Several distillation methods in our comparison are formulated for single-label classification using softmax-normalized logits, and extending them to the multi-label setting would require redesigning their original distillation formulations. We therefore adopt a controlled single-label setting so that the comparison primarily reflects differences among the CMKD methods rather than differences in their multi-label extensions. Specifically, we first remove four extremely rare genres, *News*, *Adult*, *Talk-Show*, and *Reality-TV*, from the genre labels of each sample, and discard samples that have no remaining genre label. We then retain only samples with exactly one remaining genre label. Finally, we remove genres with fewer than 50 single-label samples, together with their samples. The resulting dataset contains 5,778 paired poster-text samples from 8 classes: *Action*, *Comedy*, *Documentary*, *Drama*, *Horror*, *Sci-Fi*, *Thriller*, and *Western*, and the number of samples per class is reported in Table 7. Each paired poster-text sample shares the same label. Because this filtering substantially changes both the size and the class distribution of the original benchmark, we do not use the official split. Instead, for each seed, we construct a stratified 60%/20%/20% split for training, validation, and testing, which is shared across all compared methods. Accordingly, the MM-IMDB results should be interpreted as a customized single-label proxy for controlled CMKD comparison, and they are not directly comparable with results reported under the standard multi-label protocol.

Preprocessing. For image and text preprocessing, we use FLAVA [30] to extract modality-specific hidden representations from movie posters and plot descriptions. For image data, each poster is processed by the FLAVA image encoder, whose output consists of a CLS token and 14×14 patch tokens. To reduce the sequence length, we average-pool the patch tokens over non-overlapping 2×2 regions while preserving the CLS token. This yields image features of size $L \times D = 50 \times 768$, where $L = 1 + 7 \times 7$ denotes the resulting token length and D denotes the feature dimension. For text data, we use the movie plot as the text input. If the plot is unavailable, we use the movie title as a fallback. The text is tokenized with a maximum length of 128 and passed through the FLAVA text encoder. We

preserve the CLS token and adaptively average-pool the remaining token representations to obtain a fixed-length text sequence. This yields text features of size $L \times D = 64 \times 768$, where L denotes the pooled token length and D denotes the feature dimension.

Model. We use lightweight sequence classifiers on top of the pre-extracted FLAVA image and text features. For both modalities, each token feature is passed through a linear projection, layer normalization, and GELU activation to obtain the intermediate feature for distillation. The intermediate feature has shape $B \times 50 \times 64$ for image data and $B \times 64 \times 64$ for text data. We use attention pooling to obtain the pooled representation, with the text attention mask used to ignore padded positions. The pooled representation is passed through layer normalization, a linear layer with GELU activation, and a final 8-way classifier. Hidden dimensions are set to 64.

A.8 NYU-Depth V2

Dataset Specification. We use NYU-Depth V2 [28] for RGB-depth semantic segmentation for indoor scenes. The dataset contains 1,449 paired RGB-depth samples captured from indoor scenes, with pixel-wise label maps of size 480×640. The semantic categories include common indoor structures and objects, such as *wall*, *floor*, *ceiling*, *bed*, *chair*, *table*, *door*, and *window*. We use 41 label indices, consisting of 40 semantic classes and one void class, and exclude the void class from evaluation. We split the dataset into training, validation, and test sets with a ratio of 60%/20%/20%.

Preprocessing. For RGB data, we first apply min-max normalization to each image, resize it to 224×224, and normalize it using ImageNet statistics [5]. The processed RGB image is passed through an ImageNet-pretrained ResNet-34 [10], and we use the layer-3 feature map as the RGB representation. For depth data, we apply min-max normalization to each depth map and replicate the single-channel depth input into three channels. We then apply the same 224×224 resizing and ImageNet normalization, and pass the result through the same ResNet-34 layer-3 feature extractor. The extracted RGB and depth features have size $D \times H \times W = 256 \times 14 \times 14$, where D denotes the channel dimension and $H \times W$ denotes the spatial resolution. The semantic label maps are kept at their original spatial resolution of 480×640 and are used as pixel-wise supervision for segmentation.

Model. We use lightweight segmentation heads on top of the pre-extracted RGB and depth features. The RGB and depth branches share the same architecture and differ only in the input modality. Each branch applies two 2D convolutional blocks to the ResNet-34 layer-3 feature map. The resulting encoder output, with shape $B \times 256 \times 14 \times 14$, is used as the intermediate feature for distillation, where B denotes batch size. For segmentation prediction, the intermediate feature is passed through an upsampling decoder. The decoder progressively upsamples the feature map and produces dense semantic logits with spatial size 480×640. The output dimension is 41, corresponding to 40 semantic classes and one void class. The model is trained with pixel-wise cross-entropy loss, where the void label is ignored. For NYU-Depth V2, where RGB and depth features are spatially aligned, we apply each feature-alignment baseline to the corresponding dense intermediate feature maps whenever possible, rather than pooling the features.

B Additional Experimental Results

B.1 Results with Standard Deviations

Tables 8 and 9 report the classification and semantic segmentation results from the main paper with standard deviations over five independent runs. The mean values are identical to those reported in the corresponding main result tables, while the standard deviations provide additional information about variability across random seeds. Tables 10, 11, and 12 further report the ablation results in Tables 3, 4, and 5 with standard deviations under the same protocol.

B.2 Statistical Significance Analysis

Following the protocol of MST-Distill [19], the data split and the teacher checkpoint vary across seeds in our main experiments. Within each seed, all methods share the same data split and teacher checkpoint, ensuring a fair comparison under identical experimental conditions. While this protocol

Table 8: Classification results. A, V, T, and I denote audio, video, text, and image modalities, respectively, and each arrow indicates the teacher-to-student distillation direction. We report mean accuracy ± standard deviation over five independent runs, with the best mean result in each column shown in bold.

Method	RAVDESS		VGG-Sound	
	A → V	V → A	A → V	V → A
w/o KD	0.9208±0.0164	0.6826±0.0189	0.5031±0.0035	0.6058±0.0086
KLD	0.9118±0.0156	0.6972±0.0674	0.5153±0.0067	0.6176±0.0090
MLLD	0.9292±0.0140	0.6889±0.0461	0.5135±0.0056	0.6122±0.0047
FitNets	0.9062±0.0188	0.7146±0.0385	0.4916±0.0105	0.5962±0.0128
RKD	0.9236±0.0132	0.7014±0.0269	0.5066±0.0068	0.6085±0.0018
CRD	0.9299±0.0199	0.7007±0.0246	0.5063±0.0032	0.6110±0.0075
OFA	0.9215±0.0183	0.6562±0.0577	0.5100±0.0093	0.6129±0.0050
MGDFR	0.9313±0.0206	0.7264±0.0090	0.5225±0.0034	0.6186±0.0084
C2KD	0.9306±0.0139	0.7021±0.0314	0.5123±0.0049	0.6154±0.0045
STXD	0.9035±0.0093	0.7250±0.0165	0.5067±0.0048	0.6093±0.0033
FDD	0.9153±0.0090	0.7055±0.0158	0.5028±0.0078	0.6063±0.0093
Ours	**0.9431**±0.0097	**0.7486**±0.0231	**0.5299**±0.0017	**0.6229**±0.0059

Method	CrisisMMD		MM-IMDB	
	T → I	I → T	T → I	I → T
w/o KD	0.5403±0.0085	0.5500±0.0050	0.6645±0.0092	0.7294±0.0159
KLD	0.5377±0.0063	0.5578±0.0049	0.6732±0.0103	0.7310±0.0079
MLLD	0.5357±0.0062	0.5562±0.0059	0.6699±0.0113	0.7320±0.0096
FitNets	0.5281±0.0202	0.5507±0.0091	0.6759±0.0101	0.7296±0.0114
RKD	0.5432±0.0058	0.5569±0.0061	0.6668±0.0117	0.7317±0.0109
CRD	0.5377±0.0110	0.5560±0.0114	0.6685±0.0109	0.7299±0.0077
OFA	0.5362±0.0080	0.5565±0.0032	0.6715±0.0150	0.7339±0.0105
MGDFR	**0.5440**±0.0103	0.5584±0.0026	0.6725±0.0105	0.7382±0.0135
C2KD	0.5410±0.0052	0.5520±0.0082	0.6670±0.0041	0.7372±0.0101
STXD	0.5421±0.0069	0.5485±0.0083	0.6761±0.0116	0.7265±0.0168
FDD	0.5382±0.0088	0.5547±0.0067	0.6670±0.0108	0.7323±0.0105
Ours	0.5433±0.0062	**0.5592**±0.0071	**0.6768**±0.0055	**0.7434**±0.0139

evaluates performance across diverse CMKD settings, the reported standard deviations include variability arising from the data split and the teacher model in addition to that arising from the distillation method itself.

Controlled Significance Test. To isolate the effect of the distillation method, we conduct controlled experiments in which the data split and the teacher checkpoint are fixed and only the remaining training randomness is varied over ten runs. We compare our method with the strongest competing method in Table 1 using a two-sided paired t-test. As shown in Table 13, our method achieves statistically significant improvements ($p < 0.05$) in five of the eight transfer directions. Two additional settings, CrisisMMD I→T and MM-IMDB T→I, show near-threshold evidence with p-values close to 0.05. In CrisisMMD T→I, MGDFR achieves a slightly higher mean, but the evidence for a reliable difference is limited, indicating that our method remains competitive in this setting.

Average Rank Analysis. We further evaluate whether the performance gains remain consistent across variations in the data split under the original varying-split protocol. For each seed, we rank all methods evaluated with the same data split and teacher checkpoint, and then average the ranks over five seeds. As shown in Table 14, our method achieves the best average rank in seven of the eight transfer directions and the second-best average rank in CrisisMMD T→I. This indicates that the proposed method remains consistently competitive across changes in both the data split and the teacher checkpoint, rather than benefiting from only a few favorable runs.

Table 9: Semantic segmentation results on NYU-Depth. Quantitative comparison for two distillation scenarios, where the first modality denotes the teacher and the second denotes the student. Results are reported as mean ± standard deviation over five independent runs.

Method	Depth → RGB			RGB → Depth		
	OA ↑	CA ↑	mIoU ↑	OA ↑	CA ↑	mIoU ↑
w/o KD	0.5582±0.0112	0.2748±0.0218	0.1850±0.0154	**0.5269**±0.0070	0.2100±0.0155	0.1366±0.0110
MLLD	0.5599±0.0122	0.2792±0.0226	0.1911±0.0169	0.5250±0.0063	0.2118±0.0150	0.1399±0.0152
CRD	0.5592±0.0101	0.2677±0.0158	0.1806±0.0108	0.5261±0.0061	0.2111±0.0125	0.1385±0.0107
OFA	0.5607±0.0119	0.2881±0.0146	0.1982±0.0107	0.5242±0.0085	0.2123±0.0125	0.1409±0.0123
MGDFR	0.5601±0.0123	0.2737±0.0196	0.1878±0.0165	0.5213±0.0099	0.2162±0.0097	0.1442±0.0083
C2KD	0.5587±0.0089	0.2771±0.0090	0.1878±0.0052	0.5214±0.0062	0.2093±0.0092	0.1396±0.0074
STXD	0.5616±0.0108	0.2786±0.0220	0.1925±0.0171	0.5231±0.0102	0.2164±0.0125	0.1425±0.0124
FDD	0.5608±0.0114	0.2744±0.0279	0.1855±0.0234	0.5252±0.0063	0.2122±0.0146	0.1403±0.0153
Ours	**0.5632**±0.0103	**0.2909**±0.0260	**0.2003**±0.0194	0.5218±0.0078	**0.2184**±0.0154	**0.1456**±0.0105

Table 10: Ablation study on Stage-3 loss components with standard deviations. "Code" denotes the student-side codebook-branch objective. "Pred-Match" denotes the codebook-induced prediction matching term. Results are reported as mean ± standard deviation over five independent runs.

Code	Pred-Match	RAVDESS		VGG-Sound	
		A → V	V → A	A → V	V → A
-	-	0.9208±0.0164	0.6826±0.0189	0.5031±0.0035	0.6058±0.0086
✓	-	0.9202±0.0260	0.7333±0.0397	0.5067±0.0029	0.6098±0.0033
-	✓	0.9285±0.0234	0.7479±0.0175	0.5294±0.0013	0.6204±0.0056
✓	✓	0.9431±0.0097	0.7486±0.0231	0.5299±0.0017	0.6229±0.0059

Code	Pred-Match	CrisisMMD		MM-IMDB	
		T → I	I → T	T → I	I → T
-	-	0.5403±0.0085	0.5500±0.0050	0.6645±0.0092	0.7294±0.0159
✓	-	0.5387±0.0129	0.5534±0.0037	0.6716±0.0037	0.7363±0.0065
-	✓	0.5367±0.0100	0.5529±0.0018	0.6770±0.0054	0.7421±0.0136
✓	✓	0.5433±0.0062	0.5592±0.0071	0.6768±0.0055	0.7434±0.0139

Significance of Loss Components. Under the same controlled setting, we examine whether the contribution of the "Code" term can be explained by run-to-run noise. As shown in Table 15, adding the "Code" term to "Pred-Match" improves the mean performance in all eight transfer directions. The improvement is statistically significant in four settings, with suggestive trends in two additional settings. "Pred-Match" is expected to produce a larger individual gain because it provides supervision more directly related to the task output, whereas the "Code" term regularizes the intermediate representation. Note that the target of "Pred-Match" is the codebook-induced teacher prediction $\mathbf{q}^{(t)}$ rather than the original teacher prediction $\mathbf{p}^{(t)}$. Both terms are therefore derived from the learned teacher codebook and provide complementary supervision at the representation and prediction levels.

Significance of Code Selection Criterion. We further evaluate whether the proposed code selection criterion provides a reliable advantage over random code selection. As shown in Table 16, the combined criterion improves the mean performance in all eight transfer directions and achieves statistically significant improvements in seven of them. The remaining MM-IMDB I→T setting also shows a positive trend ($p = 0.0659$). This indicates that the proposed criterion identifies codes that are systematically more useful for cross-modal distillation than random selection.

B.3 Computational Cost Analysis

We analyze the training cost of the proposed method on VGG-Sound in terms of end-to-end wall-clock training time, peak GPU memory usage, and floating-point operations (FLOPs). For FLOPs, we exclude the teacher and student backbone forward passes, which are largely shared across methods, and report only the additional scoring or alignment operations introduced by each method. As shown in Table 17, our method requires the longest wall-clock training time because of its three-

Table 11: Ablation on the prediction matching target in Stage-3 loss with standard deviations. Results are reported as mean ± standard deviation over five independent runs.

Pred-Match Target	RAVDESS		VGG-Sound	
	A → V	V → A	A → V	V → A
Original branch ($\mathbf{p}^{(t)}$)	0.8903±0.0140	0.6201±0.0788	0.5173±0.0024	0.6171±0.0079
Codebook branch ($\mathbf{q}^{(t)}$)	0.9431±0.0097	0.7486±0.0231	0.5299±0.0017	0.6229±0.0059

Pred-Match Target	CrisisMMD		MM-IMDB	
	T → I	I → T	T → I	I → T
Original branch ($\mathbf{p}^{(t)}$)	0.5425±0.0081	0.5539±0.0046	0.6654±0.0158	0.7351±0.0139
Codebook branch ($\mathbf{q}^{(t)}$)	0.5433±0.0062	0.5592±0.0071	0.6768±0.0055	0.7434±0.0139

Table 12: Ablation study on code selection criteria with standard deviations. Results are reported as mean ± standard deviation over five independent runs.

Selection Criteria	RAVDESS		VGG-Sound	
	A → V	V → A	A → V	V → A
Random	0.9292±0.0134	0.7396±0.0214	0.5252±0.0049	0.6211±0.0051
Task relevance only	0.9396±0.0063	0.7361±0.0134	0.5276±0.0055	0.6233±0.0063
Student compatibility only	0.9375±0.0078	0.7417±0.0124	0.5275±0.0077	0.6222±0.0052
Combination (Ours)	0.9431±0.0097	0.7486±0.0231	0.5299±0.0017	0.6229±0.0059

Selection Criteria	CrisisMMD		MM-IMDB	
	T → I	I → T	T → I	I → T
Random	0.5353±0.0076	0.5536±0.0043	0.6718±0.0135	0.7353±0.0116
Task relevance only	0.5435±0.0047	0.5558±0.0072	0.6727±0.0139	0.7351±0.0104
Student compatibility only	0.5437±0.0052	0.5565±0.0058	0.6722±0.0138	0.7346±0.0103
Combination (Ours)	0.5433±0.0062	0.5592±0.0071	0.6768±0.0055	0.7434±0.0139

stage training procedure. Other multi-stage or computationally intensive methods, such as CRD, FitNets, and MGDFR, also require substantially longer training time than the student trained without distillation. However, the peak memory usage and alignment-related FLOPs of our method are not consistently higher than those of the compared methods. For instance, our method requires less peak memory and fewer alignment-related FLOPs than MGDFR. The leave-one-code-out scoring in Stage-2 accounts for only 7.9% and 7.7% of the total training time for V→A and A→V, respectively, indicating that it is not the dominant computational cost under the default configuration.

Effect of Codebook Size. Table 18 reports the stage-wise training time and accuracy under different codebook sizes B. The runtime of Stage-2 increases approximately linearly with B, whereas the runtimes of Stage-1 and Stage-3 remain nearly constant. Performance is generally robust to B, and an intermediate codebook size ($B = 512$, our default) provides the best trade-off between accuracy and computational cost. We also observe that code utilization decreases for oversized codebooks. With $B = 2048$, more than 30% of the codes frequently remain unused, and $B = 1024$ yields approximately 10-15% unused codes, whereas $B \leq 512$ generally maintains 99-100% code utilization. This suggests that excessively large codebooks increase the scoring cost while introducing redundant or underutilized codes without consistent performance gains. The code utilization measured in Stage-1 can therefore serve as a practical diagnostic for identifying unnecessarily large codebooks. Nevertheless, since the cost of Stage-2 scales linearly with B, it may become a bottleneck for substantially larger codebooks, and more efficient importance estimation remains a direction for future work.

Inference Cost. All additional components, including the teacher, the codebook, the projection layers, and the codebook branches, as well as the Stage-2 scoring procedure, are used only during training. At inference time, they are entirely removed, and only the original student encoder and task

Table 13: Controlled significance tests against the runner-up methods for the classification results. The data split and the teacher checkpoint are fixed, and results are reported as mean ± standard deviation over ten runs. p-values are computed using a two-sided paired t-test. The runner-up is STXD for MM-IMDB T→I and MGDFR for all other settings.

Method	RAVDESS		VGG-Sound	
	A → V	V → A	A → V	V → A
Runner-up	0.9271±0.0080	0.7351±0.0144	0.5219±0.0046	0.6126±0.0043
Ours	0.9420±0.0071	0.7556±0.0120	0.5273±0.0041	0.6158±0.0029
p-value	0.001	0.007	0.028	0.007

Method	CrisisMMD		MM-IMDB	
	T → I	I → T	T → I	I → T
Runner-up	0.5486±0.0062	0.5550±0.0034	0.6680±0.0048	0.7443±0.0057
Ours	0.5448±0.0084	0.5581±0.0019	0.6724±0.0037	0.7482±0.0051
p-value	0.105	0.060	0.055	0.046

Table 14: Average rank of classification methods across five runs with varying data splits and teacher checkpoints. Lower values indicate better performance, and the best average rank in each column is shown in bold.

Method	RAVDESS		VGG-Sound		CrisisMMD		MM-IMDB	
	A → V	V → A	A → V	V → A	T → I	I → T	T → I	I → T
KLD	7.4	6.4	4.6	3.8	7.4	5.4	**4.0**	7.0
MLLD	4.8	7.2	4.4	5.8	8.6	5.0	5.8	6.4
FitNets	7.8	5.8	10.8	10.8	7.4	7.8	4.4	7.4
RKD	5.4	7.0	8.0	8.2	4.4	4.6	7.6	6.6
CRD	4.6	7.4	7.8	7.8	6.2	4.8	5.8	7.0
OFA	5.8	8.8	5.6	5.0	7.2	5.2	6.0	5.6
MGDFR	4.0	4.0	2.0	2.8	**3.4**	3.8	5.8	3.6
C2KD	4.4	6.6	4.8	4.2	5.2	7.6	8.0	3.6
STXD	10.2	3.8	7.8	7.8	5.0	9.8	4.6	8.8
FDD	7.0	5.8	9.0	8.0	6.4	7.2	8.4	6.4
Ours	**2.4**	**2.0**	**1.0**	**1.4**	3.8	**3.4**	**4.0**	**2.0**

head are retained. Therefore, the proposed method introduces no additional parameters, memory, or computation at inference compared with the student trained without distillation, and the teacher modality is not required at deployment.

B.4 Discussion on the Code Retention Ratio

The proposed method retains a fixed ratio of teacher codes for student-side quantization, which is set to 50% for all classification experiments and 12.5% for all segmentation experiments. These values were coarsely chosen from a small set of candidate values and then fixed across all settings of each task to avoid extensive scenario-specific tuning. They should therefore be regarded as practical default operating points rather than individually optimized settings.

Fig. 3 shows that the best ratio varies across datasets and transfer directions, indicating that the appropriate amount of retained teacher-side information depends on the distillation scenario. A smaller ratio may discard useful teacher-side information, whereas a larger ratio may include codes that are less relevant to the task or less compatible with the student modality. Nevertheless, performance remains relatively stable over a broad range of ratios and generally exceeds the w/o KD baseline. This indicates that the effectiveness of the proposed method does not depend on a single carefully tuned ratio.

Table 15: Controlled significance tests for the contribution of the "Code" term. The data split and the teacher checkpoint are fixed, and results are reported as mean ± standard deviation over ten runs. p-values are computed using a two-sided paired t-test.

Code	Pred-Match	RAVDESS		VGG-Sound	
		A → V	V → A	A → V	V → A
-	✓	0.9274±0.0110	0.7469±0.0060	0.5246±0.0031	0.6129±0.0031
✓	✓	0.9420±0.0071	0.7556±0.0120	0.5273±0.0041	0.6158±0.0029
p-value		0.0031	0.0885	0.1215	0.0504

Code	Pred-Match	CrisisMMD		MM-IMDB	
		T → I	I → T	T → I	I → T
-	✓	0.5377±0.0057	0.5518±0.0081	0.6690±0.0045	0.7448±0.0028
✓	✓	0.5448±0.0084	0.5581±0.0019	0.6724±0.0037	0.7482±0.0051
p-value		0.0223	0.0348	0.0116	0.1302

Table 16: Controlled significance tests for the code selection criterion. The data split and the teacher checkpoint are fixed, and results are reported as mean ± standard deviation over ten runs. p-values are computed using a two-sided paired t-test.

Selection Criteria	RAVDESS		VGG-Sound	
	A → V	V → A	A → V	V → A
Random	0.9316±0.0100	0.7423±0.0073	0.5242±0.0029	0.6125±0.0023
Combination (Ours)	0.9420±0.0071	0.7556±0.0120	0.5273±0.0041	0.6158±0.0029
p-value	0.0333	0.0205	0.0309	0.0287

Selection Criteria	CrisisMMD		MM-IMDB	
	T → I	I → T	T → I	I → T
Random	0.5339±0.0074	0.5527±0.0061	0.6668±0.0043	0.7443±0.0024
Combination (Ours)	0.5448±0.0084	0.5581±0.0019	0.6724±0.0037	0.7482±0.0051
p-value	0.0105	0.0147	0.0034	0.0659

Although the fixed ratio provides consistent improvements, it does not adapt to the varying amount of transferable information across cross-modal scenarios. Adaptive code retention, which automatically determines the number of retained codes for each scenario, is therefore a promising direction for future work.

B.5 Class-wise Analysis on NYU-Depth V2

As shown in Table 2, the OA of our method in the RGB→Depth setting is slightly lower than that of the student trained without distillation (0.5218 vs. 0.5269). To examine whether this decrease reflects negative transfer, we analyze the class-wise recall difference between our method and the student trained without distillation. Table 19 reports the five classes with the largest recall increases and decreases.

The analysis shows that the decrease in OA is strongly influenced by class imbalance rather than by a general degradation across classes. The recall of `wall` decreases by 0.0274, and since this class occupies 25.6% of all pixels, this change substantially affects the pixel-weighted OA. In contrast, several less frequent classes, including `other_furniture`, `mirror`, `television`, `desk`, and `curtain`, show much larger recall improvements of 0.13-0.18, although their contributions to OA are limited by their low pixel frequencies. Accordingly, CA, which assigns equal importance to each class, improves from 0.2100 to 0.2184 despite the slight decrease in OA, and mIoU also improves from 0.1366 to 0.1456.

Table 17: Training computational costs on VGG-Sound. Time and Mem denote the end-to-end wall-clock training time (s) and peak GPU memory usage (MB), respectively. TFLOPs are measured only for the method-specific scoring and alignment operations, excluding the teacher and student forward passes that are largely shared across methods. ↪ Stage-2 denotes the cost of the leave-one-code-out scoring in our method.

Method	V → A			A → V		
	Time	Mem	TFLOPs	Time	Mem	TFLOPs
w/o KD	82.69	25.40	–	168.44	77.25	–
KLD	172.95	76.29	0.62	176.04	80.34	0.62
MSE	171.78	76.35	0.62	165.68	80.40	0.62
CRD	325.45	88.22	0.63	324.03	94.71	0.63
FitNets	338.17	78.26	1.24	339.19	83.05	1.24
MGDFR	327.07	178.07	76.81	316.99	177.95	784.21
Ours	458.85	122.86	31.56	452.29	103.65	148.21
↪ Stage-2	36.24	95.69	31.56	34.87	93.18	148.21

Table 18: Stage-wise training time (s) and accuracy under different codebook sizes B on VGG-Sound. Accuracy is averaged over five independent runs. Our default setting is $B = 512$.

B	V → A				A → V			
	Stage-1	Stage-2	Stage-3	Acc.	Stage-1	Stage-2	Stage-3	Acc.
128	231.53	10.48	182.34	0.6202	224.55	10.56	179.26	0.5270
256	228.37	18.54	190.97	0.6213	225.65	18.65	193.02	0.5272
512	226.81	36.24	195.80	0.6229	225.70	34.87	191.72	0.5299
1024	226.90	79.93	191.29	0.6201	224.92	68.62	190.45	0.5283
2048	228.29	184.83	202.42	0.6205	221.61	131.78	199.46	0.5262

These results indicate that the proposed method improves class-balanced performance, and that the OA decrease reflects a class-specific trade-off rather than systematic negative transfer across classes. The gains on several less frequent classes are also consistent with the benefit of complementary semantic cues provided by the RGB teacher.

B.6 Hard and Soft Code Assignment

We use a discrete codebook because it provides an explicit and structure-independent concept interface between heterogeneous teacher and student features. Mapping each feature unit to a finite set of codes allows us to evaluate the contribution of individual concepts, rank them, and retain only the transferable subset. Such code-level selection is less directly defined when each feature is represented by an unrestricted continuous mixture of anchors.

To examine whether hard assignment to the selected codes is necessary during transfer, we replace the nearest-code assignment in Stage-3 with a soft assignment over the same learned and selected codes. Specifically, each student unit is represented by a similarity-weighted combination of the selected codes rather than by its single nearest code. As shown in Table 20, the two variants achieve comparable performance, with hard assignment yielding slightly higher mean accuracy in both transfer directions. This result shows that the selected codes remain effective under soft assignment, while hard assignment is already sufficient in the current setting.

B.7 Code Reuse Across Student Units

The proposed method does not enforce an exclusive one-to-one correspondence between student units and teacher codes. Multiple student units can be assigned to the same code, allowing a concept to be represented across several temporal or spatial locations. To quantify this behavior, we measure the proportion of activated codes that are assigned to more than one student unit over the VGG-Sound test set. We observe that 94.48% and 97.47% of the activated codes are reused by multiple

Table 19: Class-wise recall analysis on NYU-Depth V2 in the RGB→Depth setting. Freq. denotes the percentage of pixels belonging to each class, and ΔRecall denotes the recall difference between our method and the student trained without distillation.

Top-5 increased classes			Top-5 decreased classes		
Class	Freq.	ΔRecall	Class	Freq.	ΔRecall
`other_furniture`	2.7%	+0.1807	`person`	0.3%	−0.1298
`mirror`	1.2%	+0.1519	`counter`	1.7%	−0.0478
`television`	0.7%	+0.1324	`bookshelf`	2.1%	−0.0398
`desk`	0.8%	+0.1315	`clothes`	0.8%	−0.0294
`curtain`	1.8%	+0.1311	`wall`	25.6%	−0.0274

Table 20: Comparison of hard and soft code assignment in Stage-3 on VGG-Sound. Results are reported as mean ± standard deviation over five independent runs.

Assignment	A → V	V → A
Hard	0.5299±0.0017	0.6229±0.0059
Soft	0.5254±0.0024	0.6208±0.0082

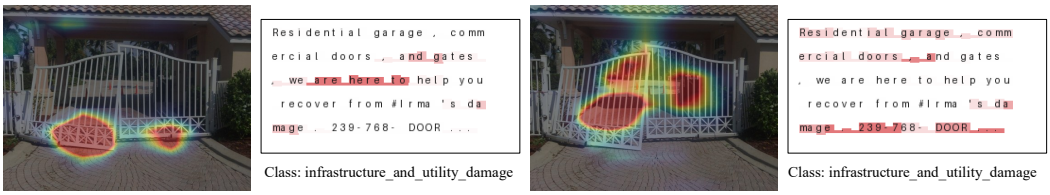
student units for A→V and V→A, respectively. This indicates that code usage is predominantly many-units-to-one-code rather than one-unit-to-one-code.

Although code assignment is performed independently for each student unit, the quantized student units retain their temporal or spatial organization and are subsequently processed jointly by the codebook-branch head. The head can therefore exploit the positions, repetition patterns, and overall arrangement of the assigned codes, which allows concepts distributed across multiple student units to be captured. However, the current assignment does not explicitly model such distributed concepts before quantization. Extending the framework to model concepts spanning multiple student units more explicitly is an interesting direction for future work.

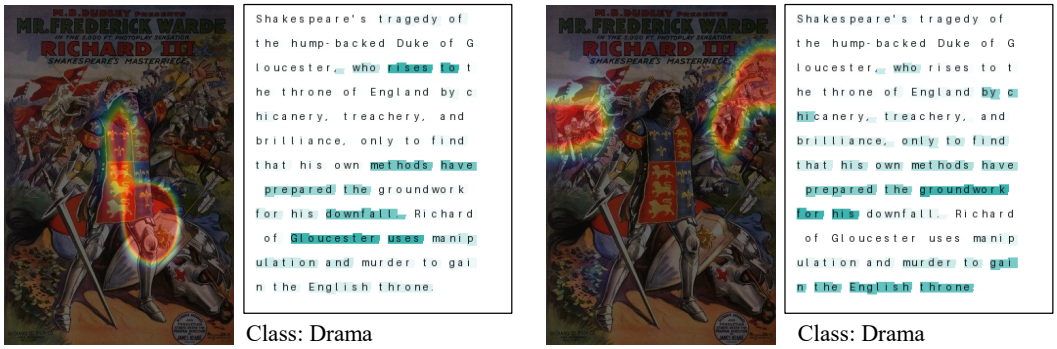
B.8 Qualitative Analysis of Learned Codes

Figs. 4a and 4b provide qualitative examples of the learned codes on CrisisMMD and MM-IMDB, respectively. We consider image-teacher distillation settings and visualize selected image-teacher codes together with high-weight tokens from the distilled text student. For each selected code, we show its associated image regions and the corresponding token weights from the text student.

In the CrisisMMD example, the sample belongs to the *infrastructure_and_utility_damage* class. The selected image-teacher codes are associated with regions around damaged door or gate structures. On the text side, the distilled student assigns relatively high weights to related textual contexts, including tokens such as *damage*, *doors*, *gates*, and *DOOR*. In the MM-IMDB example, the sample belongs to the *Drama* class. The selected codes attend to visually salient regions in the movie poster, while the text student places relatively high weights on narrative phrases related to dramatic content, such as *tragedy*, *methods*, *prepared the groundwork*, *downfall*, and *English throne*. These examples suggest that the learned codes can be associated with semantically related visual and textual cues, providing qualitative support for code-based cross-modal transfer.



(a) CrisisMMD



(b) MM-IMDB

Figure 4: Qualitative analysis of learned codes. For each selected image-teacher code, the pair shows its associated image region and corresponding high-weight tokens from the distilled text student.

C Pseudo Algorithm of The Proposed Framework

Algorithm 1 provides the pseudo-code of the proposed framework. The procedure consists of three stages, corresponding to the method description in Sec. 3: teacher-side codebook learning, code-wise importance estimation, and selective codebook-based distillation. Although the pseudo-code is written at the dataset level for clarity, all optimization steps are implemented with mini-batch training.

D Broader Impact

This work studies a general cross-modal knowledge distillation framework and is not tied to a specific deployment scenario. A potential positive impact is that the method may improve efficient model learning by transferring useful knowledge from auxiliary modalities to compact or single-modality students. However, as with other distillation methods, the student may inherit biases, spurious correlations, or undesirable behavior from the teacher model and training data. Therefore, deployment in sensitive applications should require careful evaluation of fairness, privacy, and robustness under the target use case.

Algorithm 1 Codebook-Guided Cross-Modal Knowledge Distillation

Require: Paired data $\{(\mathbf{x}_i^{(t)}, \mathbf{x}_i^{(s)}, \mathbf{y}_i)\}_{i=1}^N$, pretrained teacher $\{f^{(t)}(\cdot;\theta^{*(t)}), g^{(t)}(\cdot;\phi^{*(t)})\}$, student $\{f^{(s)}(\cdot;\theta^{(s)}), g^{(s)}(\cdot;\phi^{(s)})\}$, codebook size B, number of selected codes K, code dimension D.

Ensure: Trained student parameter $\theta^{(s)}, \phi^{(s)}$

Stage-1: Teacher-side codebook learning

Freeze the pretrained teacher parameters $\theta^{*(t)}, \phi^{*(t)}$.
Initialize codebook $\mathbf{C} \in \mathbb{R}^{B\times D}$, projections $\Pi_{\text{in}}^{(t)}, \Pi_{\text{out}}^{(t)}$, and codebook branch head $g_{\text{cp}}^{(t)}(\cdot;\phi_{\text{cp}}^{(t)})$.
for each training iteration **do**
  for $i = 1$ to N **do**
    Extract teacher features $\mathbf{z}_i^{(t)} = f^{(t)}(\mathbf{x}_i^{(t)};\theta^{*(t)})$.
    Quantize teacher features with $\mathbf{C}$ and compute codebook-induced prediction $\mathbf{q}_i^{(t)}$.
    Compute student prediction $\mathbf{p}_i^{(s)}$.
  end for
  Compute loss according to Eq. 6.
  Update $\mathbf{C}, \Pi_{\text{in}}^{(t)}, \Pi_{\text{out}}^{(t)}, \phi_{\text{cp}}^{(t)}, \theta^{(s)}, \phi^{(s)}$.
end for

Stage-2: Code-wise importance estimation

for each code $k \in \{1, \ldots, B\}$ **do**
  Remove the k-th code to construct $\mathbf{C}_{-k}$.
  Compute leave-one-code-out teacher prediction $\bar{\mathbf{q}}_i^{(t)}$.
  Compute full codebook teacher prediction $\mathbf{q}_i^{(t)}$.
  Compute student prediction $\mathbf{p}_i^{(s)}$.
  Estimate task relevance u_k^{rel} and student compatibility u_k^{comp} according to Eq. 8.
  Compute final importance score $u_k = u_k^{\text{rel}} + u_k^{\text{comp}}$.
end for
Select the top-K codes according to $\{u_k\}_{k=1}^B$ and form $\mathbf{C}_{\text{sel}}$.

Stage-3: Selective codebook-based distillation

Initialize student-side projections $\Pi_{\text{in}}^{(s)}, \Pi_{\text{out}}^{(s)}$ and student-side codebook branch head $g_{\text{cp}}^{(s)}(\cdot;\phi_{\text{cp}}^{(s)})$.
Re-initialize student model parameters $\theta^{(s)}, \phi^{(s)}$.
Fix the selected teacher codebook $\mathbf{C}_{\text{sel}}$.
for each training iteration **do**
  for $i = 1$ to N **do**
    Extract student features $\mathbf{z}_i^{(s)} = f^{(s)}(\mathbf{x}_i^{(s)};\theta^{(s)})$.
    Quantize student features with $\mathbf{C}_{\text{sel}}$ and compute student-side codebook-induced prediction $\mathbf{q}_i^{(s)}$.
    Compute teacher-side codebook-induced prediction $\mathbf{q}_i^{(t)}$.
    Compute student task prediction $\mathbf{p}_i^{(s)}$.
  end for
  Compute loss according to Eq. 12.
  Update the student model parameters $\theta^{(s)}, \phi^{(s)}$.
end for
return Trained student parameter $\theta^{(s)}, \phi^{(s)}, \Pi_{\text{in}}^{(s)}, \Pi_{\text{out}}^{(s)}, \phi_{\text{cp}}^{(s)}$.

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: The abstract and introduction clearly state the paper's main scope and contributions.

   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 paper discusses limitations and future directions in the conclusion section.

   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: This paper does not include theoretical results.

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: The paper provides the proposed algorithm in pseudo-code and reports the necessary experimental details to support reproducibility of the main results.

   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: [No]

Justification: The paper uses publicly available datasets and provides experimental details for reproducibility, but open-source code is not included in the submission.

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: The paper specifies the experimental details needed to understand the results.

   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: For the main experiments, we conduct repeated runs and report the corresponding mean and standard deviation.

   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 paper reports the computing environment used for the experiments.

   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: This work studies a general cross-modal knowledge distillation framework and is not tied to a specific deployment scenario. A potential positive impact is that the method may improve efficient model learning by transferring useful knowledge from auxiliary modalities to compact or single-modality students. However, as with other distillation methods, the student may inherit biases, spurious correlations, or undesirable behavior from the teacher model and training data. Therefore, deployment in sensitive applications should require careful evaluation of fairness, privacy, and robustness under the target use case.

    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:

    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: LLMs were not used as an important, original, or non-standard component of the core methodology. We only used LLM-based tools for paper writing, editing, formatting, and code-assistance purposes.

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.