A. Additional Experimental Settings and Results

A.1. Datasets

As briefly discussed in §4.1, we evaluate our method across five datasets: HMDB51 [21], UCF101 [39], Kinetics-600 [20], HAC [9], and EPIC-Kitchens [6].

1) **HMDB51** [21] is a video action recognition dataset containing 6,766 video clips across 51 action categories. The clips are sourced from various media, including digitized movies and YouTube videos, and include both video and optical flow modalities.

2) **UCF101** [39] is a diverse video action recognition dataset collected from YouTube, containing 13,320 clips representing 101 actions. This dataset includes variations in camera motion, object appearance, scale, pose, viewpoint, and background conditions. It provides video and optical flow modalities.

3) **Kinetics-600** [20] is a large-scale action recognition dataset with approximately 480,000 video clips across 600 action categories. Each clip is a 10-second snippet of an annotated action moment sourced from YouTube. Following [10], we selected a subset of 229 classes from Kinetics-600 to avoid potential overlaps with other datasets, resulting in 57,205 video clips. Video and audio modalities are available, with optical flow extracted at 24 frames per second using the TV-L1 algorithm [49], yielding 114,410 optical flow samples.

4) **HAC** [9] includes seven actions—such as 'sleeping', 'watching TV', 'eating', and 'running'—performed by humans, animals, and cartoon characters, with 3,381 total video clips. The dataset provides video, optical flow, and audio modalities.

5) **EPIC-Kitchens** [6] is a large-scale egocentric video dataset collected from 32 participants in their kitchens as they captured routine activities. For our experiments, we use a subset from the Multimodal Domain Adaptation paper [35], which contains 4,871 video clips across the eight most common actions in participant P22's sequence ('put,' 'take,' 'open,' 'close,' 'wash,' 'cut,' 'mix,' and 'pour'). The available modalities include video, optical flow, and audio.

A.2. Tasks

As briefly discussed in §4.2, we evaluate our method on two tasks: Near-OOD detection, and Far-OOD detection [10].

For Near-OOD detection, we evaluate using four datasets. In EPIC-Kitchens 4/4, a subset of the EPIC-Kitchens dataset is divided into four classes for training as ID and four classes for testing as OOD, totaling 4,871 video clips. HMDB51 25/26 and UCF101 50/51 are similarly derived from HMDB51 [21] and UCF101 [39], containing 6,766 and 13,320 video clips, respectively. For Kinetics-600 129/100, a subset of 229 classes is selected from Kinetics-600 [20], with approximately 250 clips per class, totaling 57,205 clips. In this setup, 129 classes are used for training (ID) and the remaining 100 for testing (OOD). We present the results of {video, optical flow} on all four datasets, and the results of {video, optical flow, audio} on Kinetics-600 dataset.

In the Far-OOD detection setup, either HMDB51 or Kinetics-600 is used as the ID dataset, with the other datasets serving as OOD datasets:

HMDB51 as ID: We designate UCF101, EPIC-Kitchens, HAC, and Kinetics-600 as OOD datasets. Samples overlapping with HMDB51 are excluded from each OOD dataset to maintain distinct ID/OOD classes. For instance, 31 classes overlapping with HMDB51 are removed from UCF101, leaving 70 OOD classes, and 8 overlapping classes are removed from EPIC-Kitchens and HAC.

Kinetics-600 as ID: We designate UCF101, EPIC-Kitchens, HAC, and HMDB51 as OOD datasets, excluding any ID class overlap with Kinetics-600. For example, 11 overlapping classes are removed from UCF101, leaving 90 OOD classes, while the original classes in EPIC-Kitchens, HAC, and HMDB51 are preserved as they have no overlap with Kinetics-600.

A.3. Baseline Design

As briefly discussed in §4.2, we compare SecDOOD against traditional on-device training classifiers combined with various post-hoc OOD detection methods. Additionally, we design two alternative baselines that do not require on-device training to further analyze the effectiveness of our approach.

For the OOD detection methods, we extend several established techniques to the multimodal setting, including MSP [15], Energy [31], MaxLogit [16], Mahalanobis [22], ReAct [40], ASH [7], GEN [32], KNN [41], and VIM [44]. These methods span multiple levels of OOD scoring, ranging from probability-based approaches (MSP, GEN), logit-based techniques (Energy, MaxLogit), and feature-space methods (Mahalanobis, KNN) to activation manipulation strategies (ReAct, ASH) and hybrid logit-feature approaches (VIM).

For the newly designed baseline methods, Ini-Classifier denotes a randomly initialized classifier that is used directly for inference without any training or fine-tuning. In contrast, Ini-Hypernetwork refers to a randomly initialized Hyper-Network that generates classifier parameters based on extracted visual features.

A.4. Implementation Details

As briefly discussed in §4.2, Near-OOD and Far-OOD tasks share the batch size of 16, the Adam optimizer, and the learning rate of 0.0001. In addition, the machine we used in the experiments is as follows:

Methods	No Mask		Mask 50% Channels		Mask 75% Channels	
	FPR95↓	AUROC↑	FPR95↓	AUROC↑	FPR95↓	AUROC↑
HMDB as ID						
Kinetics	21.89	94.29	15.39	96.06	43.79	89.97
UCF	52.79	81.93	46.18	86.19	68.76	74.61
HAC	29.42	94.02	22.35	94.41	29.76	93.91
Kinetics as ID						
HMDB	69.67	78.09	69.52	79.25	70.19	75.66
UCF	69.63	72.98	69.34	75.37	70.89	72.84
HAC	68.01	76.81	58.32	84.98	69.36	76.29

Table A. Far-OOD Detection results using various mask proportions (↑ higher is better; ↓ lower is better).

GPU server with AMD EPYC Milan 7763, 64×16 = 1TB DDR4 memory, 15 TB SSD, 6× NVIDIA RTX A6000 Ada.

The encryption time was measured on a MacBook Pro with M1 Pro.

For the Near-OOD tasks and for Far-OOD tasks with HMDB as the ID dataset (excluding Kinetics), the hypernetwork is configured as a single layer—there is no hidden layer. The weight matrix W of size (num feature × num class) and the bias b of size (num class) are directly derived from the input features, with a batch normalization applied before the parameter output.

For the Near-OOD and Far-OOD tasks with Kinetics as the ID dataset, the hypernetwork is configured with two layers, where the hidden layer dimension is set to 3584 for Near-OOD and 2048 for Far-OOD. In this case, the parameters undergo batch normalization at the latent variable stage before output; the weight matrix W of size (num feature × num class) and the bias b of size (num class) are the results after the second batch normalization.

A.5. Additional Results

As discussed in §4.4, we present additional experimental results to further analyze the impact of different masking percentages.

Tab. A summarizes the model's performance under various masking conditions. Interestingly, we observe that applying a 50% masking ratio results in only a slight performance degradation. This finding suggests that our dynamic channel sampling approach effectively removes noisy or redundant channels while maintaining the model's overall capability. We attribute this robustness to the efficiency of our sampling strategy, which dynamically selects the most informative channels, thereby mitigating the negative impact of extensive masking. Furthermore, this result provides additional evidence for the resilience of the hypernetwork, as it demonstrates that the model does not rely on highly specific input features to perform well.

Datasets	Size	4G: 5MB/s	4G: 15MB/s	5G: 50MB/s	5G: 100MB/s
Near-OOD					
HMDB	↑: 0.18 MB ↓: 4.75 MB	↑: 144 ms ↓: 950 ms	↑: 48 ms ↓: 317 ms	↑: 14 ms ↓: 95 ms	↑: 7 ms ↓: 48 ms
UCF	↑: 0.18 MB ↓: 9.40 MB	↑: 144 ms ↓: 1880 ms	↑: 48 ms ↓: 627 ms	↑: 14 ms ↓: 188 ms	↑: 7 ms ↓: 94 ms
Kinetics	↑: 0.18 MB ↓: 24.17 MB	↑: 144 ms ↓: 4834 ms	↑: 48 ms ↓: 1611 ms	↑: 14 ms ↓: 483 ms	↑: 7 ms ↓: 242 ms
EPIC	↑: 0.18 MB ↓: 0.79 MB	↑: 144 ms ↓: 158 ms	↑: 48 ms ↓: 53 ms	↑: 14 ms ↓: 16 ms	↑: 7 ms ↓: 8 ms
Far-OOD					
HMDB	↑: 0.18 MB ↓: 8.08 MB	↑: 144 ms ↓: 1616 ms	↑: 48 ms ↓: 539 ms	↑: 14 ms ↓: 162 ms	↑: 7 ms ↓: 81 ms
Kinetics	↑: 0.18 MB ↓: 42.81 MB	↑: 144 ms ↓: 8562 ms	↑: 48 ms ↓: 2854 ms	↑: 14 ms ↓: 856 ms	↑: 7 ms ↓: 428 ms

Table B. Time delay with updated encryption/decryption method and separated upload/download speeds. (↑) denotes upload from device to cloud, (↓) denotes download from cloud to device. The upload speed is assumed to be 1/4 of the download speed.

Tab. B presents the communication latency for different datasets under various network conditions, considering an updated encryption/decryption method. The upload speed is assumed to be one-fourth of the download speed, reflecting typical mobile network conditions. The results show that 4G networks with a 5MB/s download speed incur the highest delays, especially for large datasets such as Kinetics, where the download time reaches 8.56 seconds. In contrast, 5G networks significantly reduce latency, with the fastest configuration (100MB/s) achieving sub-500ms downloads for most cases. The encryption/decryption overhead is minimal, as observed in the slight increase in total transmission time. These findings highlight the benefits of high-speed networks for cloud-assisted OOD detection, particularly in handling large datasets efficiently.