EgoDistill: Egocentric Head Motion Distillation for Efficient Video Understanding

Shuhan Tan[1] Tushar Nagarajan[1] Kristen Grauman[1,2]
[1]The University of Texas at Austin [2]FAIR, Meta AI
{shuhan,tushar.nagarajan,grauman}@cs.utexas.edu

Abstract

Recent advances in egocentric video understanding models are promising, but their heavy computational expense is a barrier for many real-world applications. To address this challenge, we propose EgoDistill, a distillation-based approach that learns to reconstruct heavy egocentric video clip features by combining the semantics from a sparse set of video frames with the head motion from lightweight IMU readings. We further devise a novel self-supervised training strategy for IMU feature learning. Our method leads to significant improvements in efficiency, requiring $200\times$ *fewer GFLOPs than equivalent video models. We demonstrate its effectiveness on the Ego4D and EPIC-Kitchens datasets, where our method outperforms state-of-the-art efficient video understanding methods.*

1. Introduction

Recent advances in augmented and virtual reality (AR/VR) technology have the potential to change the way people interact with the digital world, much like the smartphone did in the previous decade. A fundamental requirement for AR/VR systems is the ability to recognize user behavior from egocentric video captured from a head-mounted camera. Towards this goal, several egocentric video datasets have been proposed in recent years, spurring increasing attention of the research community [11, 26, 56].

Recent advances in egocentric action recognition, anticipation, and retrieval focus on building powerful *clip-based* video models that operate on video clips of a few seconds at a time [12, 16, 18, 25, 43, 44, 54, 55]. Despite encouraging performance, these models typically process densely-sampled frames with temporally-aware operations, making them computationally heavy. This makes them impractical for AR/VR devices with constrained resources, or for real-time video applications that require low latency. How to efficiently perform egocentric video understanding is therefore an important, yet unsolved problem.

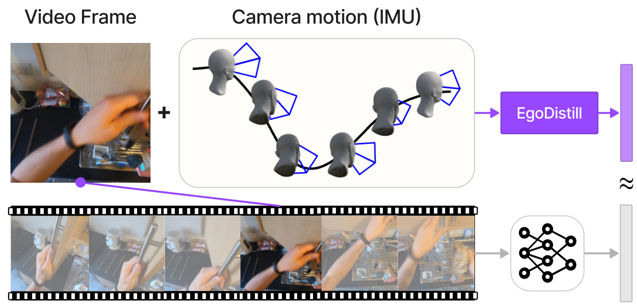

Figure 1. **Illustration of EgoDistill.** Given a single video frame and camera motion from IMU, EgoDistill learns to reconstruct the more expensive dense video clip feature. With its lightweight input, EgoDistill significantly improves efficiency.

To address this issue, we take inspiration from how animals perceive the world with ego-motion. Neuroscience research has found that during active movement, the animal visual cortex receives and utilizes head motion signals from the motor cortex for visual processing [27,52,53]. This indicates that head motion signals support an embodied agent's efficient understanding of the egocentric visual stream. Inspired by this phenomenon, we explore the relationship between human head motion and egocentric video for efficient video understanding. In practice, we consider head motion signals captured by the inertial measurement unit (IMU) of a head-mounted camera. IMU measures motion from an accelerometer and gyroscope and is widely available on popular wearable devices. Prior work leverages IMU as an extra modality for human action recognition [13, 68, 69], (e.g., jumping, walking, standing) or as geometric cues for visual-inertial odometry [7, 20, 71].

In contrast, we propose to achieve efficient video understanding by drawing on IMU as a *substitute* for dense video frame observations. The intuition is as follows. A video clip contains two things: semantic content (appearance of objects, places, people) and dynamics (how the scene and the camera move). While densely sampled frames are sure to capture both of the above—as done by current clip models [16, 17, 54]—we hypothesize they are sometimes overkill. For a short video clip, much of the semantic content is intelligible from even a single frame; meanwhile, the head motion provides a good portion of the dynamics, implicitly revealing how the visual appearance changes across neighboring frames.

Building on this insight, we introduce EgoDistill, an approach that learns to reconstruct dense egocentric video clip features using temporally sparse visual observations (as few as one RGB frame) together with the head motion from IMU. Specifically, EgoDistill employs a new form of knowledge distillation from video models. During training, we train a lightweight model that takes sparsely sampled image(s) and IMU to approximate the video representation extracted by a powerful but expensive video model. We further improve the model with a novel self-supervised training stage for IMU feature learning. During inference, we directly utilize the lightweight model for egocentric video recognition, leading to much higher efficiency. Our model is flexible to the target heavy video feature, as we demonstrate with multiple current leading egocentric video models [16–18, 54]. See Figure 1.

Importantly, EgoDistill offers a major efficiency gain: processing low-dimensional IMU and a few frames is much more efficient compared to processing a dense stack of frames. In practice, EgoDistill uses 200× fewer GFLOPs than the original video model.

We experiment on the largest available egocentric action recognition datasets: Ego4D [26] and EPIC-Kitchens-100 [11]. We show that IMU coupled with an image offers better cross-modality knowledge distillation performance than images alone or images with audio. For a typical 50-minute egocentric video, EgoDistill reduces inference time of the source video model from 25 minutes to *36 seconds*. Moreover, with only 1-4 frames, our lightweight distillation model achieves a better accuracy-efficiency trade-off than state-of-the-art models for adaptively sampling video content [50, 65]. Notably, we surpass the accuracy of these fast approaches by a large margin while requiring 4× to 8× less computation.

2. Related Work

IMU for activity recognition. Recent work explores using the IMU sensor on mobile devices for human activity recognition of actions like walking, jumping, or sitting [3, 47, 59–61]. Normally, these models take input from IMU sensors mounted on human body joints [9, 42, 60], waist-mounted [41] or in-pocket smartphones [33]. See [64] for a survey. Abundant work in video recognition explores ways to learn from RGB coupled with other modalities—audio [1, 22, 38], optical flow [19, 57, 58] or both [32, 36, 51]—but comparatively fewer use IMU [48], and unlike our work, they focus on third-person video [13, 68, 69] and do not target at model efficiency. Our idea is for IMU to help reconstruct more expensive video features, rather than simply fuse IMU with RGB for multi-modal recognition.

IMU for odometry. Inertial odometry aims to estimate the position and orientation of the camera-wearer with readings from the IMU sensor. Traditionally, methods rely on IMU double integration [4] or enhancements thereof [5, 35, 40]. Recent data-driven methods automatically learn to perform inertial odometry with supervised [30, 70] or self-supervised learning [7], or combine IMU and visual input for more robust estimates with visual-inertial odometry [20, 71]. While IMU can convey geometric ego-motion to our learned model, our goal is to produce efficient egocentric video features rather than to output odometry.

Visual feature learning with IMU. IMU is also used to learn better vision features [14,15,34,63], e.g., to encourage image features that are equivariant with ego-motion [34], to predict an IMU-captured body part (leg, hand) [14, 15], or to predict video-IMU correspondence [63], for applications like action recognition [15, 63] and scene understanding [14, 34]. While these results reinforce that IMU can inject embodied motion into visual features, our idea to use head motion to infer pretrained video features for speedy video understanding is distinct.

Efficient video recognition. Being crucial for mobile applications, efficient video recognition has received increasing attention in recent years. Several studies focus on designing lightweight architectures [17, 30, 37, 62, 72] by reducing 3D CNN operations across densely sampled frames. Our idea is orthogonal to them as we focus on inputs with sparsely-sampled frames. As we show in experiments, our method is compatible with different video architectures.

Another line of research achieves efficiency by adaptively selecting video content to process. Some reduce *temporal redundancy* by adaptively selecting which video clip [39], frames [24, 49], and/or feature channel [50] to process and which to skip, while others reduce *spatial redundancy*, efficient recognition by dynamically selecting selecting for each frame a smaller but important region to process [66, 67]. Other work dynamically selects tokens in video transformers among both the spatial and temporal dimensions [65]. Our idea is complementary: rather than dynamically subsample the available video content, we show how to *infer "full" video features for every clip* using static image(s) and motion data. Our results outperform state-of-the-art sampling models (cf. Sec. 4). In addition, we focus on egocentric video, where head motion is particularly meaningful for inferring unobserved visual content. To our knowledge, ours is the first technique specifically aimed at accelerating egocentric video processing.

Multimodal distillation. Knowledge distillation aims to transfer knowledge learned by an expensive model to a lightweight model [31]. Recent work explores multimodal distillation, e.g., transferring from a RGB model to a flow or depth model [23, 28], from a 3D model to a 2D model [45], or from a visual model to audio model [2, 21]. The ListenToLook model [22] incorporates both clip subsampling and video-to-audio distillation for fast activity recognition in third-person video. In contrast, we explore the relationship between the camera-wearer's head motion and RGB signals for egocentric video. Our experiments show EgoDistill's advantage over ListenToLook in terms of the speed-accuracy tradeoff on egocentric video datasets.

3. Approach

We introduce EgoDistill, which uses sparsely-sampled frames and head motion from IMU to approximate the features of heavy video models for efficient egocentric video understanding. We first introduce the egocentric action recognition task (Sec. 3.1). Then, we introduce our pipeline (Sec. 3.2), our distillation model and training objective (Sec. 3.3), and our self-supervised IMU feature learning (Sec. 3.4). Figure 2 overviews our approach.

3.1. Egocentric action recognition

Given a fixed-length video clip $\mathcal{V} \in \mathbb{R}^{T\times H\times W\times 3}$ consisting of T RGB frames of size $H\times W$ and a set of C action classes, the task of action recognition is to output a score for each action class, representing its likelihood. Typically, this is done with a powerful but expensive video model Ω, that directly operates on all the available frames to output the C class logits $\Omega(\mathcal{V}) \in \mathbb{R}^C$. Ω is trained with standard classification loss:

$$\mathcal{L}_{\text{ACT}} = \sum_{\mathcal{V}_i} \mathcal{L}_{\text{CE}}(c_i, \sigma(\Omega(\mathcal{V}_i))), \quad (1)$$

where $\mathcal{V}_i$ is the i-th video clip in the dataset, c_i is the corresponding ground-truth action label, σ is the softmax function, and $\mathcal{L}_{\text{CE}}$ is cross-entropy loss. Popular video recognition models use clips that are typically ~2 seconds long [16, 18, 54]. For longer videos, scores are averaged across all clips it contains to infer the video action label.

3.2. Efficient video inference with head motion

Processing the video clip $\mathcal{V}$ for action recognition is computationally intensive; however, the computation cost can be modulated depending on how frames from the clip are used. On the one hand, *clip-based* models [16–18, 54] process most (or all) frames in a video clip $\mathcal{V}$ to achieve strong recognition performance, but come at a high computational cost. On the other hand, *frame-level* models [24, 49, 51, 67] only process one (or a small number) of frames from $\mathcal{V}$ and are more efficient, but suffer a drop in performance as a result. Our goal is to train a frame-based model that can approximate heavy clip-based model performance while maintaining high efficiency.

For this, we turn to head motion captured by IMU. Along with RGB frames, each video clip is paired with IMU measurements $\mathcal{M}$ that record the camera (head) motion during the video. Specifically, the IMU readings are composed of 6-dimensional accelerometer and gyroscope measurements in the xyz axes, which encode strong temporal motion information about camera pose changes (both translation and rotation) across frames.

For short video clips, a set of sparsely sampled frames $\mathcal{I}$ often already captures most *appearance* information. Complementary to this, the IMU readings capture *camera motion* information (see below for discussion on scene motion). Moreover, IMU is very efficient to process due to its low dimensionality. By processing inputs from these two sources with a lightweight frame-based model, we can infer the semantic and dynamic features of a heavier clip-based video model.

Given $\mathcal{I}$ and $\mathcal{M}$, we train an efficient lightweight model Φ to approximate the output of video model Ω. Specifically, we train our EgoDistill model Φ that achieves

$$\Phi(\mathcal{I}, \mathcal{M}) \approx \Omega(\mathcal{V}). \quad (2)$$

Such a lightweight model will be able to approximate the result of the heavy video model, while being much more efficient. Our approach is agnostic to the specific video model Ω; in experiments, we demonstrate its versatility for MotionFormer [54], MViT [16], SlowFast [18] and X3D [17].

In practice, we uniformly sample N frames[1] from $\mathcal{V}$ to obtain $\mathcal{I}$. We can achieve a trade-off between efficiency and performance by changing the number of frames N. In our experiments we use very low values of N (1 to 4 frames). In the next section, we discuss how we train Φ.

3.3. Video feature distillation with IMU

We achieve the objective in Equation 2 via knowledge distillation [31], where we transfer knowledge learned by the expensive teacher model Ω to a lightweight student model Φ. Next we present the design of Φ and the training objectives, followed by our self-supervised IMU feature pretraining stage in Sec. 3.4.

We design Φ to be a two-stream model. For a video clip and associated IMU signal $(\mathcal{I}, \mathcal{M})$, we extract image features $\mathbf{z}_\mathcal{I} = f_\mathcal{I}(\mathcal{I})$ and IMU features $\mathbf{z}_\mathcal{M} = f_\mathcal{M}(\mathcal{M})$ using lightweight feature encoders $f_\mathcal{I}$, $f_\mathcal{M}$ respectively. Then, we fuse $\mathbf{z}_\mathcal{I}$ and $\mathbf{z}_\mathcal{M}$ with a fusion network Π to obtain the fused VisIMU feature $\mathbf{z}_\phi = \Pi(\mathbf{z}_\mathcal{I}, \mathbf{z}_\mathcal{M})$. Finally, a fully-connected layer uses the fused feature to predict class logits $\Phi(\mathcal{I}, \mathcal{M}) \in \mathbb{R}^C$.

The fused feature $\mathbf{z}_\phi$ contains semantic information from the image frame coupled with complementary motion infor-

[1]Other frame sampling heuristics (e.g., selecting from the start or center of the video) performed equivalently or worse than uniform sampling.

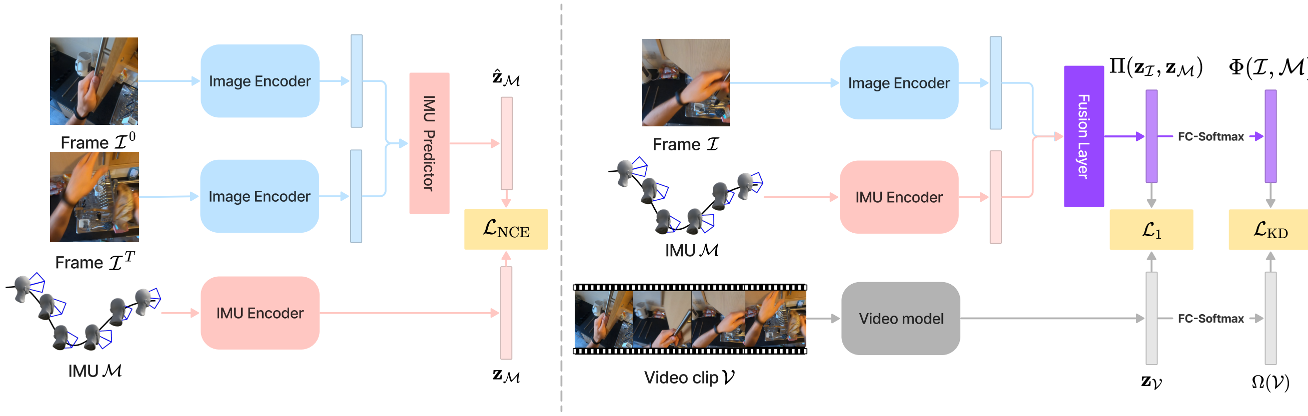

Figure 2. **EgoDistill architecture. Left: Self-supervised IMU feature learning.** Given start and end frames of a clip, we train the IMU encoder to anticipate visual changes. **Right: Video feature distillation with IMU.** Given image frame(s) and IMU, along with our pretrained IMU encoder, our method trains a lightweight model with knowledge distillation to reconstruct the features from a heavier video model. When the input includes more than one image frame, the image encoder aggregates frame features temporally with a GRU.

mation from IMU, allowing us to accurately reconstruct the video clip feature. See Figure 2.

We train Φ with a combination of three losses, as follows. First, we train Φ to approximate the original video feature $\mathbf{z}_{\mathcal{V}}$ from the video model Ω:

$$\mathcal{L}_1 = \sum_{(\mathbf{z}_{\mathcal{V}_i}, \mathbf{z}_{\phi_i})} \left\| \mathbf{z}_{\mathcal{V}_i} - \mathbf{z}_{\phi_i} \right\|_1 . \tag{3}$$

This cross-modal loss encourages the fused feature $\mathbf{z}_\phi$ to match the video feature, i.e., the combined features from the different modalities should match in the feature space.

Training with $\mathcal{L}_1$ alone does not fully capture the classification output of Ω. Therefore, we also train Φ with a knowledge distillation loss:

$$\mathcal{L}_{\text{KD}} = \sum_{(\mathcal{V}_i, \mathcal{I}_i, \mathcal{M}_i)} \mathcal{D}_{\text{KL}}(\sigma(\Omega(\mathcal{V}_i)/\tau), \sigma(\Phi(\mathcal{I}_i, \mathcal{M}_i)/\tau)), \tag{4}$$

where $(\mathcal{V}_i, \mathcal{I}_i, \mathcal{M}_i)$ represents the i-th clip in the dataset, $\mathcal{D}_{\text{KL}}$ measures KL-divergence between the class logits from the teacher model Ω and student model Φ, and τ is a temperature parameter. Intuitively, $\mathcal{L}_{\text{KD}}$ casts the output of the video teacher model as a soft target for training the student model. In this way, the student model learns to better generalize by mimicking the output distribution of the heavy video model.

Finally, to further encourage the features to preserve elements useful for activity understanding, we also compute an action classification loss:

$$\mathcal{L}_{\text{GT}} = \sum_{(\mathcal{I}_i, \mathcal{M}_i)} \mathcal{L}_{\text{CE}}(c_i, \sigma(\Phi(\mathcal{I}_i, \mathcal{M}_i))), \tag{5}$$

where c_i is the ground-truth action label, following Equation 1. The final training loss is a combination of these three loss functions:

$$\mathcal{L} = \alpha \mathcal{L}_{\text{KD}} + (1 - \alpha)\mathcal{L}_{\text{GT}} + \beta \mathcal{L}_1, \tag{6}$$

where α controls the balance between knowledge distillation and activity training [31], and β controls the weight for feature space matching.

Critically, processing a few image frame(s) and the low-dimensional IMU readings is substantially faster than processing the entire video. Once trained, our model can approximate the behavior of the source video model for recognition tasks, with the key benefit of efficient egocentric video recognition.

What kind of motion does our model preserve? Video motion decomposes into *scene* motion (e.g., how the objects and the camera wearer's hands are moving on their own), and *camera* motion (i.e., how the camera wearer is moving their head). By itself, IMU would directly account only for camera motion, not scene motion. However, by learning to map from the RGB frame *and* IMU to the *full* video feature, we are able to encode predictable scene motions tied to scene content, e.g., how does hand and object movement in subsequent frames relate to the camera wearer's head motion (see Figure 7). Moreover, our model is applied to relatively short clips (1-2 seconds) in sequence, which means the appearance content is regularly refreshed as we slide down to process the longer video.

3.4. Self-supervised IMU feature learning

The success of EgoDistill depends on how well the IMU feature encoder $f_{\mathcal{M}}$ extracts useful camera motion information and associates it with the visual appearance change in the video clip. In this way EgoDistill can learn to anticipate unseen visual changes in the video with $\mathcal{I}$ and $\mathcal{M}$. We design a self-supervised pretraining task to initialize the weights of $f_{\mathcal{M}}$ to achieve this.

Specifically, for each clip $\mathcal{V}$, we obtain its first and last frames $(\mathcal{I}^0, \mathcal{I}^T)$ as well as the IMU $\mathcal{M}$. We first extract visual features $\mathbf{z}_{\mathcal{I}}^0, \mathbf{z}_{\mathcal{I}}^T$ and IMU feature $\mathbf{z}_{\mathcal{M}}$ with feature extractors $f_{\mathcal{I}}$ and $f_{\mathcal{M}}$ mentioned above. Then, we train a feature predictor h to predict the IMU feature $\hat{\mathbf{z}}_{\mathcal{M}} = h(\mathbf{z}_{\mathcal{I}}^0, \mathbf{z}_{\mathcal{I}}^T)$. By connecting $\hat{\mathbf{z}}_{\mathcal{M}}$—which is a function of image features only—with $\mathbf{z}_{\mathcal{M}}$, we encourage $f_{\mathcal{M}}$ to extract useful camera motion features specifically associated with the visual appearance changes. Note that those appearance changes may include scene motion. Therefore, we include an $\mathcal{L}_1$ loss to train $f_{\mathcal{M}}$, which encourages $f_{\mathcal{M}}$ to extract motion features accounting for scene motion in the full video.

In sum, we train $f_{\mathcal{M}}$, h, and the fusion network Π using $\mathcal{L}_1$ and NCE loss [29]: $\mathcal{L}_{\text{pretrain}} = \mathcal{L}_{\text{NCE}} + \mathcal{L}_1$, where

$$\mathcal{L}_{\text{NCE}} = \sum_i -\log \frac{\text{sim}(\hat{\mathbf{z}}_{\mathcal{M}_i}, \mathbf{z}_{\mathcal{M}_i})}{\sum_j \text{sim}(\hat{\mathbf{z}}_{\mathcal{M}_i}, \mathbf{z}_{\mathcal{M}_j})}. \quad (7)$$

We sample negative examples $\mathbf{z}_{\mathcal{M}_j}$ from other instances in the same mini-batch for $j \neq i$, and $\text{sim}(q, k) = \exp(\frac{q \cdot k}{|q||k|} \frac{1}{\tau'})$ with temperature $\tau' = 0.1$[2].

To summarize, prior to the main training stage of Equation 6, we pretrain the IMU feature extractor $f_{\mathcal{M}}$ and fusion network Π. As we will show below, both pretraining losses result in IMU features that are consistent with visual changes and lead to better finetuning performance.

4. Experiments

We evaluate our approach for resource-efficient action recognition.

4.1. Experimental setup

Datasets. We experiment on two large-scale egocentric action recognition datasets. **Ego4D** [26] contains 3,670 hours of egocentric videos of people performing diverse tasks (from cooking to farming) across the globe. As action recognition is not part of the original Ego4D benchmark, we construct this task with annotations from the Hands+Objects temporal localization benchmark [26] (see Supp. for details). We include clips with paired IMU and audio[3], and consider classes with at least 2 labeled instances. This results in a 94-class action recognition dataset with 8.5k training videos and 3.6k evaluation videos. **EPIC-Kitchens** [11] contains 100 hours of egocentric videos capturing daily activities in kitchen environments. We use annotations from the action recognition benchmark. Similar to Ego4D, we select videos that have paired IMU and audio data, and split the resulting data by camera-wearer. This results in a 62-class action dataset with 29k training videos and 6.2k evaluation videos. For both datasets, we use "verb" labels as the target for action recognition as they are well aligned to activity motions.

Evaluation metrics. To measure action recognition performance, we report the per-video top-1 accuracy on the validation set. We densely sample clips from each video and average their predictions to compute accuracy. To benchmark efficiency, we measure computational cost with FLOPs (floating-point operations) during inference.

Implementation details. In our main experiments, we use MotionFormer [54] as the video teacher model Ω due to its strong performance for egocentric video. For EPIC-Kitchens, we use the authors' provided checkpoint. For Ego4D, we finetune the above model for 50 epochs with $1e^{-4}$ learning rate and 64 batch size on the training set. We use 16-frame input with sample rate 4. For the student model Φ, we use a ResNet-18 as the image backbone $f_{\mathcal{I}}$ and a 1D Dilated CNN [6] for the IMU backbone $f_{\mathcal{M}}$. The feature fusion module Π uses a concatenation operation following a two-layer fully-connected layer with hidden dimension 1024. For each video clip, the input image(s) is resized to 224×224, and the IMU is a 422×6 matrix (around 2 seconds with 198Hz frequency), representing the accelerometer and gyroscope readings along the xyz axes. For the image input, we uniformly sample N frames from the video clip. If $N > 1$, we use $f_{\mathcal{I}}$ to sequentially generate features for each frame and aggregate them with a GRU module [10]. For both datasets, we first pretrain the model with the self-supervised objective (Section 3.4) for 50 epochs with AdamW [46] using batch size 64 and learning rate $1e^{-4}$. Then, we finetune all the models with the same setting (Equation 6). We set $\alpha = 0.95$ and $\beta = 1.0$ based on validation data. For Ego4D, we set $\tau = 10.0$ and train the model for 150 epochs. For EPIC-Kitchens, we set $\tau = 1.0$ and train for 50 epochs.

4.2. Baselines

We compare to the following methods:

- **AdaFuse** [50] trains a lightweight policy network to adaptively compute (or skip) feature map channels for each frame during inference. We use the AdaFuse$^{\text{TSN}}_{\text{R50}}$ model with the provided hyper-parameters.
- **STTS** [65] trains a module to rank spatio-temporal tokens derived from videos in a transformer-based model, and selects only the top-K tokens to speed up inference.
- **ListenToLook** [22]: uses the audio-based feature distillation module from [22] following the same audio processing and model architecture.

These methods represent recent advances in efficient video recognition models. AdaFuse represents state-of-the-

[2] We keep the ImageNet-pretrained $f_{\mathcal{I}}$ model frozen, as finetuning it leads to mode collapse.

[3] We require audio to compare with the audio-based baseline [22].

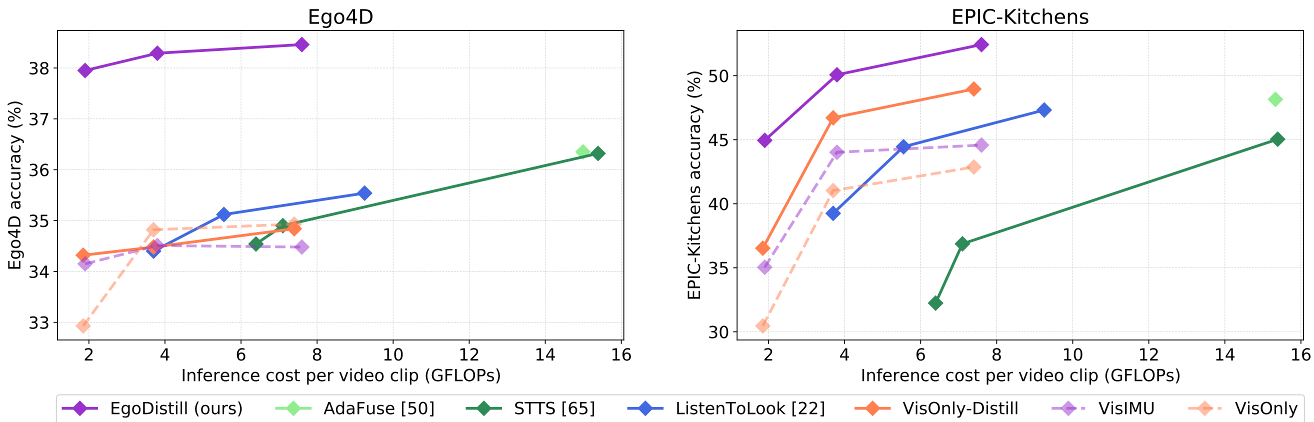

Figure 3. **Accuracy vs. efficiency for action recognition on Ego4D (left) and EPIC-Kitchens (right).** EgoDistill outperforms state-of-the-art efficient video recognition methods that adaptively sample video content, while using 4× to 8× fewer GFLOPs.

$\mathcal{L}_{\text{KD}}$	$\mathcal{L}_1$	$\mathcal{L}_{\text{GT}}$	$\mathcal{L}_1$-pretrain	$\mathcal{L}_{\text{NCE}}$-pretrain	Ego4D	EPIC-Kitchens
		✓			34.15	35.04
	✓	✓	✓	✓	35.51	39.33
✓		✓	✓	✓	37.71	42.20
✓	✓		✓	✓	37.46	43.17
✓	✓	✓			36.99	41.21
✓	✓	✓		✓	37.26	42.30
✓	✓	✓	✓		37.49	43.51
✓	✓	✓	✓	✓	**37.95**	**44.95**

Table 1. **Ablation study of model components.** We compare the accuracy of EgoDistill with different components under $N = 1$.

Source Model	Ego4D			EPIC-Kitchens		
	Video	EgoDistill	VisOnly-D	Video	EgoDistill	VisOnly-D
MFormer [54]	46.38	**37.95**	34.32	77.28	**44.95**	37.20
MViT [16]	40.32	**36.46**	33.40	53.38	**36.90**	31.22
SlowFast [18]	40.52	**33.29**	33.04	58.34	**39.42**	33.47
X3D [17]	37.56	**33.57**	32.90	52.28	**36.34**	31.71

Table 2. **Versatility to model architectures.** EgoDistill outperforms the baseline for multiple common architectures, showing the generality of our idea. "Video" refers to the more expensive source model. We show the model accuracy under $N = 1$.

art approaches that achieve efficiency by reducing temporal redundancy in CNN models. STTS is one of the most recent approaches that efficiently reduces both spatial and temporal redundancy in ViT models, which achieves the state-of-the-art on Kinectics-400 [8]. ListenToLook also relies on distillation, but using audio rather than head motion. For each model we generate multiple versions with different computation budgets to plot accuracy vs. GFLOPs. We train all AdaFuse and STTS models with 4 input frames to align with the maximum frames used by our model. For AdaFuse, we use the only provided hyper-parameter in the paper.[4] For STTS, we use three provided variants: $\text{T}^0_{0.5}$-$\text{S}^4_{0.7}$, $\text{T}^0_{0.8}$-$\text{S}^4_{0.9}$ and the full model without token selection. For ListenToLook we adopt the same efficiency-accuracy trade-off as our method, i.e., varying the number of input frames.

In addition, we test variants of our method:

- **VisOnly-Distill** is our model without the IMU branch and fusion layer but trained with the same loss function. Performance of this model reveals the role of IMU in the process of distillation.

- **VisIMU** is our model trained with only $\mathcal{L}_{\text{GT}}$ in Equation 5. It shows the effectiveness of distillation from the video model compared with directly training the features with action labels.

- **VisOnly** is an image-only model trained with $\mathcal{L}_{\text{GT}}$, which serves as the baseline.

[4]Modifying hyper-parameters to control the accuracy-efficiency trade-off results in unstable training and unreliable performance.

4.3. Main Results

Importance of IMU-guided distillation. Figure 3 shows the accuracy vs. efficiency curves. Methods towards the top-left of the plot represent those with both high accuracy and efficiency. Our method achieves good accuracy with low computational cost. Specifically, on EPIC-Kitchens, when $N = 1$, EgoDistill improves over VisOnly-Distill by 8.4% with only a small increase in computation. This result shows the effectiveness of IMU for reconstructing egocentric video features. Compared to VisIMU, EgoDistill improves by 9.9%, showing the effectiveness of knowledge distillation from the video model. Importantly, this reveals that EgoDistill does not simply benefit from the extra IMU context; our idea to approximate video features is necessary for best results. We see similar results on Ego4D.

Comparison with the state of the art. Figure 3 also shows that EgoDistill achieves better accuracy with less computation than existing efficient video recognition models AdaFuse [50], STTS [65], and ListenToLook [22].

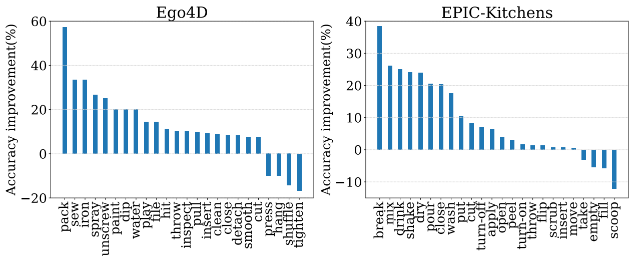

Figure 4. **Per-class accuracy improvement** over VisOnly-Distill. Best and worst performing classes are shown.

	GFLOPs	Runtime (ms)	Parameters (M)
Video [54]	369.51	10.70	108.91
AdaFuse [50]	15.20	2.04	38.85
STTS [65]	7.19	1.63	36.63
ListenToLook [22]	3.10	0.43	25.53
EgoDistill	**1.91**	**0.25**	**20.56**

Table 3. **Efficiency analysis.** Our approach is the most efficient. "Video" refers to the original (full-clip) feature. Lower is better.

With $N = 4$ frames, EgoDistill surpasses STTS by 7.4% and AdaFuse by 4.2% on EPIC-Kitchens, with 2× fewer GFLOPs, and surpasses both methods by 2.1% on Ego4D. In addition, EgoDistill surpasses ListenToLook by 7.4% and 2.9% on EPIC-Kitchens and Ego4D respectively, which suggests that head motion is more informative than audio for feature reconstruction in egocentric video.

4.4. Analysis

Model component ablations. Table 1 ablates different design choices in our model, setting $N = 1$ for all experiments. We observe that training EgoDistill without $\mathcal{L}_1$, $\mathcal{L}_{\mathrm{KD}}$ or $\mathcal{L}_{\mathrm{GT}}$ deteriorates performance. Specifically, training without $\mathcal{L}_{\mathrm{KD}}$ leads to the largest performance drop, which indicates that knowledge distillation is an essential component in our approach. Training without $\mathcal{L}_1$ also leads to a significant performance drop, which shows the importance of our idea to align features from the different modalities. Further, our self-supervised pretraining stage is very effective at training the IMU extractor to encode useful motion information that is consistent with visual feature change. Finally, we compare with a model that simply does multi-modal recognition with IMU (top row). The strong contrast here indicates the importance of our idea to use IMU to predict video model features, as opposed to simply adding IMU as an additional input modality.

Impact of teacher video model architecture. In our main experiments we use MotionFormer [54] as the teacher video model due to its strong performance on egocentric video tasks. To emphasize the generality of our idea, we show the performance of EgoDistill with other video teacher architectures in Table 2. Similar to the MotionFormer model, we train these models on each of the labeled datasets, and then train our model using the resulting video models as the teacher. As expected, better video teacher models lead to better student model performance. More importantly, we observe consistent improvement by EgoDistill over the VisOnly-Distill baseline on both datasets and with different video teacher models, highlighting our idea's generality and versatility.

Figure 5. **Best (top) and worst (bottom) reconstructed videos.**

Where does our model work best/worst? In Figure 3 we saw that using IMU leads to an overall performance improvement on action recognition, indicating better video feature prediction capability. Next, we explore what kinds of clips are better reconstructed using EgoDistill. Figure 4 shows the improvement of EgoDistill over the VisOnly-Distill model on Ego4D and EPIC-Kitchens split by action class. We observe that IMU is more useful for actions with predictable head motion (e.g., *break*, *cut*, *close*), and is less helpful for actions where head motion may be small or unrelated (e.g., *empty*, *fill*, *press*).

Figure 5 shows clip examples whose video features are best and worst reconstructed. We observe that the best reconstructed clips (top) contain moderate head motion that is predictive of scene motion and action semantics. For example, the camera wearer's head moves slightly backwards while opening the cabinet. On the other hand, more poorly reconstructed clips tend to contain little head motion (third row)—in which case IMU is redundant to the RGB frame—or drastic head motion that is weakly correlated with the camera wearer's activity and introduces blur to the frame (last row).

Efficiency analysis. To compare the efficiency of different models, aside from GFLOPs, we also compare their inference run-time and number of parameters. For run-time, we record the time spent to infer a single video clip's label with a single A40 GPU, and take the average time over the full validation datasets of Ego4D and EPIC-Kitchens with batch-size of 32. Table 3 shows the results. EgoDistill runs much faster than the other methods. Notably, it reduces the GFLOPs of MotionFormer by nearly 200×. Furthermore,

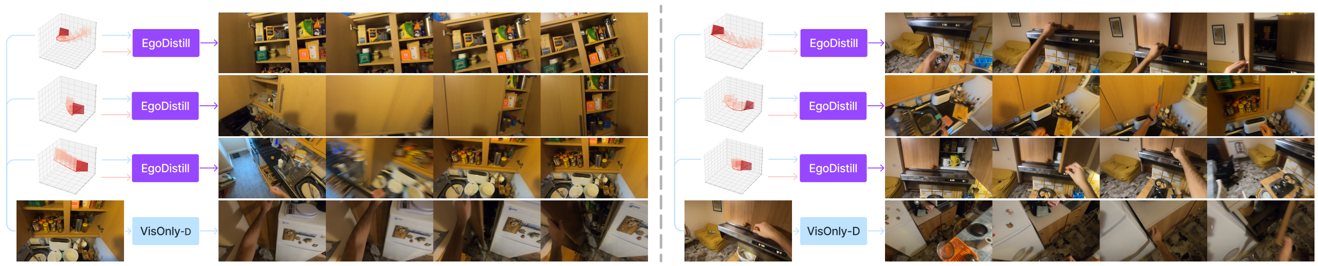

Figure 6. **Retrieving video clips with EgoDistill.** Given a query frame (bottom left) and a paired IMU segment (red camera frustums) , we retrieve the nearest clip in the video dataset according to EgoDistill and visualize its (unobserved) frames (strip to the right). Compared to VisOnly-Distill, which outputs a single feature for a given input frame (bottom row), EgoDistill outputs a distinct feature by conditioning on IMU, showing its ability to preserve both semantic and motion during reconstruction. For instance, in the top-right example, EgoDistill retains the cabinet interaction semantics in the frame as well as the upward camera-motion in the IMU. Zoom in to view best.

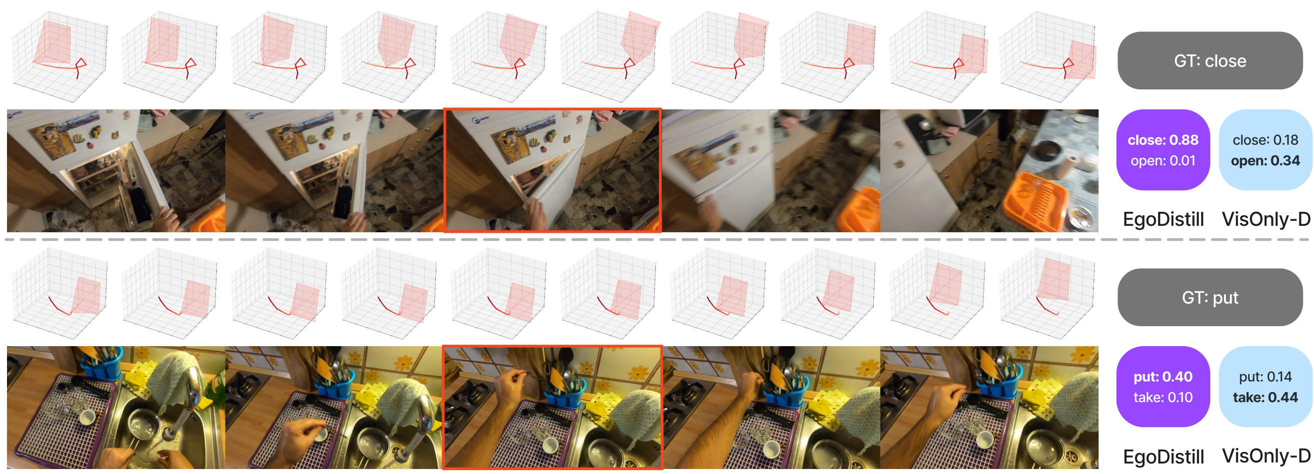

Figure 7. **Anticipating scene motion with EgoDistill.** For each clip, we show the head motion and video frames. Note, only the center frame (red border) is observed by the model. Action classification scores are shown on the right. EgoDistill can successfully anticipate scene motion and disambiguate the action semantics in the input frame. For example, in the top center frame, the image alone cannot reveal if the door is being opened or closed, whereas our feature, learned with head motion, recovers correlations with the *scene motion* (i.e., hand motion and door motion) to disambiguate "close" from "open". A similar effect for "put" vs. "take" is seen in the second example.

it runs $6.5\times$ faster than STTS [65] while achieving 4.4% higher accuracy on EPIC-Kitchens.

4.5. Qualitative Results

What do EgoDistill features capture? To explore this, we pair a single input frame with different IMU clips as inputs to EgoDistill, then retrieve the nearest video clip for each resulting anticipated video feature. Figure 6 illustrates this. We see that EgoDistill outputs video features that all involve interaction with the cabinet (right panel), and is able to use different IMU inputs to retrieve different video clips that show consistent camera motion. In contrast, VisOnly-Distill only retains the semantic context to retrieve a single clip. These results indicate that EgoDistill is able to approximate video features that capture both semantic and motion information. See Supp. for more (and animated) results.

Is there evidence EgoDistill captures scene motion? Figure 7 shows how our features learned with *head* motion can nonetheless expose certain scene motion cues. EgoDistill improves the accuracy over VisOnly-Distill on ambiguous categories (like *close* and *put*) by a large margin (20.3% and 10.4% on EPIC-Kitchens, 8.5% and 3.9% on Ego4D). See caption for details.

5. Conclusion

We present EgoDistill, the first model to explore egocentric video feature approximation for fast recognition. Experiments on action recognition on Ego4D and EPIC-Kitchens demonstrate that our model achieves a good balance between accuracy and efficiency, outperforming state-of-the-art efficient video understanding methods. Our approach has great potential to accelerate video understanding for egocentric videos using a data stream that is already ubiquitous in egocentric cameras. In the future, we plan to investigate how to use head motion for long-term human activity understanding with room context and visual correspondence learning for multi-view videos.

References

[1] Relja Arandjelović and Andrew Zisserman. Look, listen and learn. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 609–617, 2017. 2

[2] Yusuf Aytar, Carl Vondrick, and Antonio Torralba. Soundnet: Learning sound representations from unlabeled video. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 892–900, Red Hook, NY, USA, 2016. Curran Associates Inc. 3

[3] Anna Borowska-Terka and Pawel Strumillo. Person independent recognition of head gestures from parametrised and raw signals recorded from inertial measurement unit. *Applied Sciences*, 10(12), 2020. 2

[4] John E. Bortz. A new mathematical formulation for strapdown inertial navigation. *IEEE Transactions on Aerospace and Electronic Systems*, AES-7:61–66, 1971. 2

[5] Agata Brajdic and Robert K. Harle. Walk detection and step counting on unconstrained smartphones. *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, 2013. 2

[6] M. Brossard, S. Bonnabel, and A. Barrau. Denoising imu gyroscopes with deep learning for open-loop attitude estimation. *IEEE Robotics and Automation Letters*, 5(3):4796–4803, 2020. 5, 12

[7] Xiya Cao, Caifa Zhou, Dandan Zeng, and Yongliang Wang. Rio: Rotation-equivariance supervised learning of robust inertial odometry. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6614–6623, June 2022. 1, 2

[8] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 6

[9] Chen Chen, Roozbeh Jafari, and Nasser Kehtarnavaz. Utd-mhad: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 168–172, 2015. 2

[10] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *EMNLP*, 2014. 5

[11] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, , Antonino Furnari, Jian Ma, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. *International Journal of Computer Vision (IJCV)*, 130:33–55, 2022. 1, 2, 5, 12

[12] Samyak Datta, Sameer Dharur, Vincent Cartillier, Ruta Desai, Mukul Khanna, Dhruv Batra, and Devi Parikh. Episodic memory question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19119–19128, June 2022. 1

[13] Alexander Diete, Timo Sztyler, and Heiner Stuckenschmidt. Vision and acceleration modalities: Partners for recognizing complex activities. In *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pages 101–106, 2019. 1, 2

[14] Kiana Ehsani, Hessam Bagherinezhad, Joseph Redmon, Roozbeh Mottaghi, and Ali Farhadi. Who let the dogs out? modeling dog behavior from visual data. In *CVPR*, 2018. 2

[15] Kiana Ehsani, Daniel Gordon, Thomas Nguyen, Roozbeh Mottaghi, and Ali Farhadi. Learning visual representation from human interactions. *International Conference on Learning Representations*, 2021. 2

[16] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *ICCV*, 2021. 1, 2, 3, 6

[17] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 200–210, 2020. 2, 3, 6

[18] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6201–6210, 2019. 1, 2, 3, 6

[19] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1933–1941, 2016. 2

[20] Christian Forster, Luca Carlone, Frank Dellaert, and Davide Scaramuzza. On-manifold preintegration for real-time visual–inertial odometry. *IEEE Transactions on Robotics*, 33(1):1–21, 2017. 1, 2

[21] C. Gan, H. Zhao, P. Chen, D. Cox, and A. Torralba. Self-supervised moving vehicle tracking with stereo sound. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7052–7061, Los Alamitos, CA, USA, nov 2019. IEEE Computer Society. 3

[22] Ruohan Gao, Tae-Hyun Oh, Kristen Grauman, and Lorenzo Torresani. Listen to look: Action recognition by previewing audio. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10457–10467, 2020. 2, 3, 5, 6, 7, 12

[23] Nuno C. Garcia, Pietro Morerio, and Vittorio Murino. Modality distillation with multiple stream networks for action recognition. In *The European Conference on Computer Vision (ECCV)*, September 2018. 3

[24] Amir Ghodrati, Babak Ehteshami Bejnordi, and Amirhossein Habibian. Frameexit: Conditional early exiting for efficient video recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 2, 3

[25] Rohit Girdhar, Mannat Singh, Nikhila Ravi, Laurens van der Maaten, Armand Joulin, and Ishan Misra. Omnivore: A Single Model for Many Visual Modalities. In *CVPR*, 2022. 1

[26] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh Kumar Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael

Wray, Mengmeng Xu, Eric Zhongcong Xu, Chen Zhao, Siddhant Bansal, Dhruv Batra, Vincent Cartillier, Sean Crane, Tien Do, Morrie Doulaty, Akshay Erapalli, Christoph Feichtenhofer, Adriano Fragomeni, Qichen Fu, Christian Fuegen, Abrham Gebreselasie, Cristina Gonzalez, James Hillis, Xuhua Huang, Yifei Huang, Wenqi Jia, Weslie Khoo, Jachym Kolar, Satwik Kottur, Anurag Kumar, Federico Landini, Chao Li, Yanghao Li, Zhenqiang Li, Karttikeya Mangalam, Raghava Modhugu, Jonathan Munro, Tullie Murrell, Takumi Nishiyasu, Will Price, Paola Ruiz Puentes, Merey Ramazanova, Leda Sari, Kiran Somasundaram, Audrey Southerland, Yusuke Sugano, Ruijie Tao, Minh Vo, Yuchen Wang, Xindi Wu, Takuma Yagi, Yunyi Zhu, Pablo Arbelaez, David Crandall, Dima Damen, Giovanni Maria Farinella, Bernard Ghanem, Vamsi Krishna Ithapu, C. V. Jawahar, Hanbyul Joo, Kris Kitani, Haizhou Li, Richard Newcombe, Aude Oliva, Hyun Soo Park, James M. Rehg, Yoichi Sato, Jianbo Shi, Mike Zheng Shou, Antonio Torralba, Lorenzo Torresani, Mingfei Yan, and Jitendra Malik. Ego4d: Around the World in 3,000 Hours of Egocentric Video. In *IEEE/CVF Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 2, 5, 12

[27] Grigori Guitchounts, Javier Masís, Steffen B.E. Wolff, and David Cox. Encoding of 3d head orienting movements in the primary visual cortex. *Neuron*, 108(3):512–525.e4, 2020. 1

[28] Saurabh Gupta, Judy Hoffman, and Jitendra Malik. Cross modal distillation for supervision transfer. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2827–2836, 2016. 3

[29] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 297–304, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR. 5

[30] Sachini Herath, Hang Yan, and Yasutaka Furukawa. Ronin: Robust neural inertial navigation in the wild: Benchmark, evaluations, & new methods. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3146–3152, 2020. 2

[31] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *NIPS Deep Learning Workshop*, abs/1503.02531, 2014. 2, 3, 4

[32] Yu Huang, Chenzhuang Du, Zihui Xue, Xuanyao Chen, Hang Zhao, and Longbo Huang. What makes multi-modal learning better than single (provably). In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. 2

[33] Andrey Ignatov. Real-time human activity recognition from accelerometer data using convolutional neural networks. *Applied Soft Computing*, 62:915–922, 2018. 2

[34] D. Jayaraman and K. Grauman. Learning image representations tied to egomotion. In *ICCV*, 2015. 2

[35] A.R. Jimenez, F. Seco, C. Prieto, and J. Guevara. A comparison of pedestrian dead-reckoning algorithms using a low-cost mems imu. In *2009 IEEE International Symposium on Intelligent Signal Processing*, pages 37–42, 2009. 2

[36] Evangelos Kazakos, Arsha Nagrani, Andrew Zisserman, and Dima Damen. Epic-fusion: Audio-visual temporal binding for egocentric action recognition. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 2

[37] Dan Kondratyuk, Liangzhe Yuan, Yandong Li, Li Zhang, Mingxing Tan, Matthew Brown, and Boqing Gong. Movinets: Mobile video networks for efficient video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16020–16030, 2021. 2

[38] Bruno Korbar, Du Tran, and Lorenzo Torresani. Cooperative learning of audio and video models from self-supervised synchronization. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 7774–7785, Red Hook, NY, USA, 2018. Curran Associates Inc. 2

[39] Bruno Korbar, Du Tran, and Lorenzo Torresani. Scsampler: Sampling salient clips from video for efficient action recognition. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6231–6241, 2019. 2

[40] Masakatsu Kourogi and Takeshi Kurata. A method of pedestrian dead reckoning for smartphones using frequency domain analysis on patterns of acceleration and angular velocity. pages 164–168, 05 2014. 2

[41] Jennifer R. Kwapisz, Gary M. Weiss, and Samuel A. Moore. Activity recognition using cell phone accelerometers. *SIGKDD Explor. Newsl.*, 12(2):74–82, mar 2011. 2

[42] Heike Leutheuser, Dominik Schuldhaus, and Bjoern M. Eskofier. Hierarchical, multi-sensor based classification of daily life activities: Comparison with state-of-the-art algorithms using a benchmark dataset. *PLOS ONE*, 8(10):1–11, 10 2013. 2

[43] Yiming Li, Ziang Cao, Andrew Liang, Benjamin Liang, Luoyao Chen, Hang Zhao, and Chen Feng. Egocentric prediction of action target in 3d. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022. 1

[44] Shaowei Liu, Subarna Tripathi, Somdeb Majumdar, and Xiaolong Wang. Joint hand motion and interaction hotspots prediction from egocentric videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3282–3292, June 2022. 1

[45] Zhengzhe Liu, Xiaojuan Qi, and Chi-Wing Fu. 3d-to-2d distillation for indoor scene parsing. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4462–4472, 2021. 3

[46] Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam, 2018. 5

[47] Niall Lyons, Avik Santra, and Ashutosh Pandey. Improved deep representation learning for human activity recognition using imu sensors. In *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 326–332, 2021. 2

[48] Sharmin Majumder and Nasser Kehtarnavaz. Vision and inertial sensing fusion for human action recognition: A review. *IEEE Sensors Journal*, 21(3):2454–2467, 2021. 2

[49] Yue Meng, Chung-Ching Lin, Rameswar Panda, Prasanna Sattigeri, Leonid Karlinsky, Aude Oliva, Kate Saenko, and

Rogerio Feris. Ar-net: Adaptive frame resolution for efficient action recognition. *arXiv preprint arXiv:2007.15796*, 2020. 2, 3

[50] Yue Meng, Rameswar Panda, Chung-Ching Lin, Prasanna Sattigeri, Leonid Karlinsky, Kate Saenko, Aude Oliva, and Rogerio Feris. Adafuse: Adaptive temporal fusion network for efficient action recognition. In *International Conference on Learning Representations*, 2021. 2, 5, 6, 7

[51] Rameswar Panda, Chun-Fu Chen, Quanfu Fan, Ximeng Sun, Kate Saenko, Aude Oliva, and Rogerio Feris. AdaMML: Adaptive Multi-Modal Learning for Efficient Video Recognition. In *International Conference on Computer Vision (ICCV)*, 2021. 2, 3

[52] Philip R.L. Parker, Morgan A. Brown, Matthew C. Smear, and Cristopher M. Niell. Movement-related signals in sensory areas: Roles in natural behavior. *Trends in Neurosciences*, 43(8):581–595, 2020. 1

[53] Philip R. L. Parker, Elliott T. T. Abe, Emmalyn S. P. Leonard, Dylan M. Martins, and Cristopher M. Niell. Joint coding of visual input and eye/head position in v1 of freely moving mice. *bioRxiv*, 2022. 1

[54] Mandela Patrick, Dylan Campbell, Yuki M. Asano, Ishan Misra Florian Metze, Christoph Feichtenhofer, Andrea Vedaldi, and João F. Henriques. Keeping your eye on the ball: Trajectory attention in video transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 1, 2, 3, 5, 6, 7

[55] Chiara Plizzari, Mirco Planamente, Gabriele Goletto, Marco Cannici, Emanuele Gusso, Matteo Matteucci, and Barbara Caputo. E2(go)motion: Motion augmented event stream for egocentric action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19935–19947, June 2022. 1

[56] Gunnar A Sigurdsson, Abhinav Gupta, Cordelia Schmid, Ali Farhadi, and Karteek Alahari. Charades-ego: A large-scale dataset of paired third and first person videos. *arXiv preprint arXiv:1804.09626*, 2018. 1

[57] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. 2

[58] Shuyang Sun, Zhanghui Kuang, Lu Sheng, Wanli Ouyang, and Wei Zhang. Optical flow guided feature: A fast and robust motion representation for video action recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[59] Yin Tang, Lei Zhang, Fuhong Min, and Jun He. Multi-scale deep feature learning for human activity recognition using wearable sensors. *IEEE Transactions on Industrial Electronics*, pages 1–1, 2022. 2

[60] Wenjin Tao, Haodong Chen, Md. Moniruzzaman, Ming C. Leu, Zhaozheng Yi, and Ruwen Qin. Attention-based sensor fusion for human activity recognition using imu signals. *ArXiv*, abs/2112.11224, 2021. 2

[61] Catherine Tong, Jinchen Ge, and Nicholas D Lane. Zero-shot learning for imu-based activity recognition using video embeddings. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 5(4):1–23, 2021. 2

[62] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018. 2

[63] Satoshi Tsutsui, Ruta Desai, and Karl Ridgeway. How you move your head tells what you do: Self-supervised video representation learning with egocentric cameras and imu sensors. *ICCV Workshop on Egocentric Perception, Interaction and Computing (EPIC)*, 2021. 2

[64] Jindong Wang, Yiqiang Chen, Shuji Hao, Xiaohui Peng, and Lisha Hu. Deep learning for sensor-based activity recognition: A survey. *Pattern Recognit. Lett.*, 119:3–11, 2019. 2

[65] Junke Wang, Xitong Yang, Hengduo Li, Liu Li, Zuxuan Wu, and Yu-Gang Jiang. Efficient video transformers with spatial-temporal token selection. In *ECCV*, 2022. 2, 5, 6, 7, 8

[66] Yulin Wang, Zhaoxi Chen, Haojun Jiang, Shiji Song, Yizeng Han, and Gao Huang. Adaptive focus for efficient video recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021. 2

[67] Yulin Wang, Yang Yue, Yuanze Lin, Haojun Jiang, Zihang Lai, Victor Kulikov, Nikita Orlov, Humphrey Shi, and Gao Huang. Adafocus v2: End-to-end training of spatial dynamic networks for video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 3

[68] Haoran Wei, Pranav Chopada, and Nasser Kehtarnavaz. C-mhad: Continuous multimodal human action dataset of simultaneous video and inertial sensing. *Sensors*, 20(10), 2020. 1, 2

[69] Haoran Wei and Nasser Kehtarnavaz. Simultaneous utilization of inertial and video sensing for action detection and recognition in continuous action streams. *IEEE Sensors Journal*, 20(11):6055–6063, 2020. 1, 2

[70] Hang Yan, Qi Shan, and Yasutaka Furukawa. Ridi: Robust imu double integration. In *ECCV*, 2018. 2

[71] Mingyu Yang, Yu Chen, and Hun-Seok Kim. Efficient deep visual and inertial odometry with adaptive visual modality selection. *ArXiv*, abs/2205.06187, 2022. 1, 2

[72] Mohammadreza Zolfaghari, Kamaljeet Singh, and Thomas Brox. ECO: efficient convolutional network for online video understanding. In *ECCV*, 2018. 2

The supplementary materials of this work consist of:

A. Supplementary video.

B. Dataset details.

C. Implementation details.

D. Additional analysis of our model.

A. Supplementary Video

In our supplementary video, we have a brief introduction of our work. More importantly, we show animated videos of Best and Worse reconstructed clips (Figure 5), Retrieving video clips with EgoDistill (Figure 6), and Anticipating scene motion with EgoDistill (Figure 7).

Animated version of these figures better show head motion and video dynamics. We recommend viewing the supplementary video for better understanding of our method and results.

B. Dataset Details.

We use two datasets in our experiments: Ego4D [26] and EPIC-Kitchens-100 [11]. In this section we describe more details about how we create our training and evaluation data.

1. **Ego4D** [26] contains 3,670 hours of egocentric videos of people performing diverse tasks (from cooking to farming) across the globe. As action recognition is not part of the original Ego4D benchmark, we construct this task with annotations from the Hands+Objects temporal localization benchmark [26]. Specifically, for each hand-objects interaction temporal annotation, we take the video clip between the pre-frame and post-frame of the annotation as input, and use the annotated verb for this interaction as label.

   We include clips with paired IMU and audio, and consider classes with at least 2 labeled instances, resulting in 94 action categories with 12.1k videos in total. In average, each clip has 2.2 second duration. Then, we randomly split data from each category into training and evaluation sets with 70%:30% ratio. Finally, we obtain a 94-class action recognition dataset with 8.5k training videos and 3.6k evaluation videos.

2. **EPIC-Kitchens** [11] contains 100 hours of egocentric videos capturing daily activities in kitchen environments. We use annotations from the action recognition benchmark in our experiment.

   We select videos that have paired IMU and audio data, and split the resulting data by camera-wearer, ensuring non-overlapping splits following the original benchmark setting. Specifically, we take videos captured by camera-wearer id starting with P30, P35, P37 as evaluation videos and use all the remaining videos as training videos. This results in a 62-class action dataset with 29k training videos and 6.2k evaluation videos.

C. Implementation Details.

IMU input processing. For each input clip, IMU input is a 422×6 matrix (around 2 seconds with 198Hz frequency), representing the accelerometer and gyroscope readings along the xyz axes. We observe that the raw IMU input has significant drifting and bias issues. This induces inconsistent correspondence between camera motion and IMU reading across different clips and videos. Therefore, for IMU reading of each clip, on each dimension we separately subtract raw readings by the mean values on the corresponding dimension. This operation normalizes IMU readings in each dimension to have zero average value. In this way, our model can only focus on the temporal motion patterns in each clip.

Audio input processing. For ListenToLook [22], we process the audio input in the same way mentioned in the paper. Specifically, we subsample the audio at 16kHZ, and compute STFT using Hann window size of 400 and hop length of 160. Please refer to [22] for more details.

Model architecture. For the image backbone, we use the ImageNet-pretrained ResNet-18 model. For the IMU backbone, we use a 5-layer 1D Dilated CNN, as found effective for IMU data processing [6]. We use the same network setting (kernel dimension, dilation gap and channel dimension) as in prior work [6]. The feature fusion model consists of a concatenation operation following two fully-connected layers with hidden dimension of 1024. Each layer except for the output layer is followed by a ReLU activation. The output dimension is the same as the teacher video model's feature dimension (768 in the case of MotionFormer). When $N > 1$, we use a one-layer GRU module to aggregate extracted features for each frame. We use a single-directioal GRU with hidden dimension of 512.

Model training. We train our models in two stages. In the self-supervised IMU feature learning stage, we train random initialized IMU encoder $f_{\mathcal{M}}$, IMU predictor h and the fusion network Π with $\mathcal{L}_{\text{NCE}}$. Here the image encoder $f_{\mathcal{I}}$ is a fixed ImageNet pretrained model. On both datasets, we train the model for 50 epochs with AdamW and batch size 64. The initial training rate is $1e^{-4}$. We decay the training rate by 0.1 at epoch 30 and epoch 40. In the second video feature distillation stage, we initialize the model with parameters obtained in the last stage and finetune. On both datasets, we use AdamW with batch size 64 and initial learning rate $1e^{-4}$. On Ego4D, we train for 150 epochs. We decay the training rate by 0.1 at epoch 90 and epoch 120. On EPIC-Kitchens, we train for 50 epochs. We decay the training rate by 0.1 at epoch 30 and epoch 40.

	Ego4D	EPIC-Kitchens
uniform	38.46	52.43
random	36.85	48.48
first	38.68	46.40
last	35.46	41.72
center	37.04	44.85

Table 4. **Effect of frame selection.** We compare the accuracy of using different frame selection heuristics for EgoDistill when $N = 4$. We observe that Uniform on average achieves better results.

D. Analysis.

Effect of frame selection. In Section 3.2, we mentioned that we use uniform sampling to obtain the N frames from each video clip. In this section, we compare the performance of our work under uniform sampling with other heuristics. Specifically, we compare with random sampling, the first N frames, the last N frames and the center N frames. We show the results in Table 4 under $N = 4$. These results indicate that uniform sampling leads to the best performance on average. Intuitively, uniform sampling on average leads to a broader coverage of both semantic contexts as well as scene motion.

Why we set N to be small. In our experiments, we set N to be 1 to 4. Using larger N (*e.g.,* 8 or 16) with densely sampled frames could lead to better results of all the methods with more computational cost. Efficient video understanding methods could benefit more as they have better temporal aggregation mechanisms given densely-sampled frames. However, the core purpose of our model is to deal with cases where we only use a few number of samples. Therefore, our model is not comparable to video clip models under dense-frame setting. Furthermore, setting N to be a small number is very important in many applications. As loading more image frames takes additional time and memory, applications with streaming videos or low-resource AR/VR devices will benefit from loading only a few frames.