Ambient AI: Multimodal Sensor Understanding

Parthiv Dupakuntla Prachi Jain Pranay Reddy Anthireddy Snehal Prabhu
College of Information and Computer Sciences
University of Massachusetts Amherst

Abstract

In this project, we explore the potential of Inertial Measurement Unit (IMU) sensor readings in egocentric videos to establish a joint contextual symmetry between text-IMU-video embeddings. Despite the shortage of IMU modeling literature, we propose an enhancement to the IMU2CLIP model through multi-objective loss optimization, normalization techniques, and the development of an IMU Summarizer. Our approach combines knowledge distillation with the baseline to outperform the standalone IMU2CLIP model. The effectiveness of our enhanced IMU Encoder is demonstrated through its superior performance on real-world downstream applications. This research opens new avenues for cross-modal applications and offers a more efficient way of performing media retrieval or activity recognition in videos.

1 Introduction

1.1 Task Description

Existing IMU2CLIP Model (16) enforces the alignment of the IMU, video, and textual representations into the joint representation space of CLIP (1). (16) introduces a novel pretraining approach for the IMU encoders which helps IMU encoder translate the IMU sensor data to provide a more intuitive representation of the user movements which is later aligned with the video data to finetune the IMU2CLIP model. This opens doors for the exploration of new and unique cross-modal applications. We evaluate the performance of our proposed enhanced IMU Encoder on the following real-world downstream applications:

Task 1. IMU Retrieval via Textual Queries (Text→IMU), where the goal is to retrieve a window of IMU signals given free-form textual queries. Once the IMU signals are retrieved, we can also retrieve the corresponding videos, allowing for a new and power-efficient way of performing media retrieval or online action detection. The retrieval performance is measured on the held-out test set using metrics such as Recall@k and Mean Reciprocal Rank (MRR), using text narrations as the queries and the IMU signals as the retrieval pool.

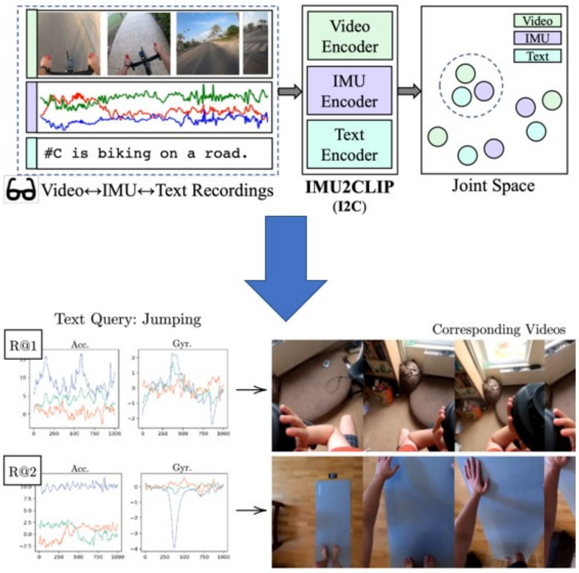

Figure 1: Illustration of IMU2CLIP (I2C) where the model aligns parallel video↔IMU↔text data in the joint space (top) and IMU-based media retrieval (bottom) as a downstream task. Given a textual query (e.g. "jumping") IMU2CLIP's predictions of the semantically closest IMU signals from the Ego4D test set (top-2) (bottom-left). Gold-parallel videos corresponding to the retrieved IMU signals. (bottom-right)

Task 2. Video Retrieval based on IMU (IMU→Video) aims to retrieve videos based on IMU signals, allowing for an intuitive way of analyzing motion signal data. We will measure the performance on the held-out test set, using the IMU signals as queries and the videos as the retrieval target.

Task 3. IMU-based Activity Recognition (IMU→Video→Text) aims to predict a discrete activity label given a window of IMU signals. We will use the soft annotations for Ego4D via text matching of the narrations provided and also explore experimenting with this task in a zero-shot setting.

1.2 Motivation

With the growing popularity of smart glasses (new-generation wearable devices), first-person or *egocentric* videos have recently become much more popular (5), (7), (13). These egocentric videos are often accompanied by parallel head-mounted Inertial Measurement Unit (IMU) sensor readings, which record device wearer's linear and rotational movements (accelerations). Given its low power consumption and low privacy implications, IMU is regarded as an essential modality for powering various on-device models that require an understanding of the device wearer's movement patterns (e.g., exercise/activity recognition for health applications). Previous works on IMU modeling typically focused on purpose-built datasets with manual annotations (12) and (4), that were limited in scale. Consequently, the utilization of IMU models in real-world scenarios has been confined to a relatively small number of use cases. Different from prior work modeling IMU in a specific task, IMU2CLIP focuses on learning general IMU representations by aligning IMU with other modalities (e.g. video and text), which can enable wider downstream applications.

2 Related Work

Image Contrastive Learning (1) introduces the Contrastive Language-Image Pretraining (CLIP) model which jointly trains the image encoder and the text encoder to predict similar pairs of image and text for the data using contrastive learning by grouping together similar pairs of images and text. (1). CLIP is pre-trained on massive image and text data for the model to learn features useful for downstream tasks. This pre-training approach helps CLIP to improve the state-of-the-art results for other image classification models. We implement this approach on the CLIP4CLIP model and further pre-train our model on a specific set of data to examine any improvement in any downstream task.

Knowledge Distillation (10) is a technique used in deep learning to transfer knowledge from a larger, more complex model (teacher model) to a smaller, more efficient model (student model). The student model is trained to mimic the behavior of the teacher model by minimizing the difference between their outputs. Distillation losses are the discrepancies between the teacher and student model outputs, which the student model aims to minimize during training. In the context of IMU2CLIP Encoder, the relevance of distillating knowledge is to train a smaller, more efficient student model that can effectively encode IMU embeddings into CLIP's latent space given textual descriptions.

The classic distillation loss (”soft target”) (10) transfers rich information by making the student model learn from the entire ”soft” output probability distribution of the teacher model instead of just learning from the 'hard' labels (the class with the highest probability). We apply different flavors of distillation losses in our experiments to investigate its effectiveness in IMU modeling inspired from their functionalities as follows. (22), (18), and (20) help preserve complex structural and relational information between aligned IMU and text embeddings. (11) aims to transfer the feature extraction capabilities by encouraging the student model to ”like” the same features as the teacher model using fewer neurons for the same information. (25) and (2) transfers rich information in the form of attention maps (features), and logits (raw/unnormalized model predictions). (19) help's student learn model uncertainty from teacher's experience. Hint-based Training (21) use intermediate representations (hints) from teacher to guide student's model training. (9) preserves the decision making process of the teacher model in the student model.

Triplet Loss is used in various contrastive learning applications, such as image recognition, object detection, and natural language processing one of the early works that used triplet loss in a contrastive learning setting is by Koch et al in (14) - ”Siamese neural networks for one-shot image recognition” which introduced the idea of using siamese neural networks and triplet loss for learning image representations that can be used for one-shot image recognition. Triplet loss was also used in recent works such as SimCLR by Chen et al (3), and in MoCo v3 by Chen et al. (2021) (8) which demonstrate its effectiveness in contrastive learning and show state-of-the-art results on various benchmarks.

3 Approach

3.1 Baseline

Existing IMU2CLIP model (16) performs cross-modal contrastive learning using InfoNCE (17) which is inspired from Noise Contrastive Estimation(NCE), an estimation method for an unnormalized probabilistic model, assigning a proxy binary classification task, where the binary task is to discriminate between data samples (positive keys) and the noise samples (negative keys). (17) proved that minimizing InfoNCE loss is equivalent to maximizing Mutual Information. The learning objective of this symmetric cross-modal contrastive loss is to maximize the joint probability or minimize the negative log-likelihood over the training set. In a nutshell, we train the baseline IMU2CLIP model in such a way that the textual descriptions or narrations resembles its corresponding IMU representation.

3.2 Distilling Knowledge in Cross-Modal Contrastive Learning.

Distilling Knowledge (*see figure: 2*) in self-supervised tasks often results in better performance than training the student model from scratch, as the student model can leverage the teacher model's knowledge and experience and thus transfer rich feature representation from the teacher. This approach has proved to be more robust and energy efficient. Distillation losses perform better

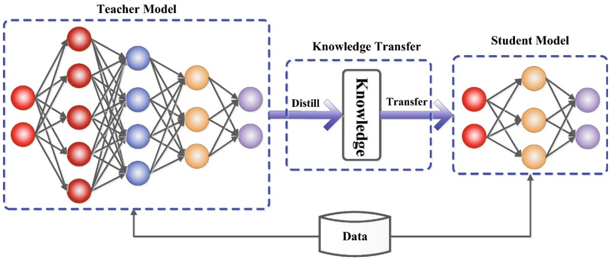

Figure 2: The teacher-student framework for Knowledge Distillation. (6)

generalizations by making the student model learn to mimic the teacher model's behavior from the seen data to unseen data.

We explore the possibility of distilling knowledge in the case of cross-modal contrastive learning. We investigate this possibility by combining contrastive loss and distillation loss as a multi-objective loss function (see equation 1) where λ_{KD} is a weight parameter that balances the two loss components. Empirically we find that performance is robust to the choice of weight parameters, and we make λ_{KD} = 1 in practice.

$$\mathcal{L}_{\text{overall}} = \mathcal{L}_{\text{InfoNCE}} + \lambda_{\text{KD}} \cdot \mathcal{L}_{\text{Distillation}} \tag{1}$$

Here $\mathcal{L}_{\text{InfoNCE}}$ is defined by equation 2 , γ is a temperature parameter that controls the concentration of the distributions and B is a batch of ground-truth IMU↔Video↔Text.

$$\mathcal{L}_{\text{InfoNCE}} = -\frac{1}{B}\sum_{i=1}^{B} \log \frac{\exp(sim(i_i, t_t))^{1/\gamma}}{\sum_{k=1}^{B} \exp(sim(i_i, t_t))^{1/\gamma}} \tag{2}$$

$\mathcal{L}_{\text{Distillation}}$ is usually defined by the vanilla distillation loss - **”soft target”** (10) as seen in equation 3. It transfers rich information in the form of ”soft” probabilities that provide insights about the relative confidence of the teacher model among different classes, which can be valuable information for the student model to learn. Thus, the student becomes more robust to variations in input data by learning from soft targets using Kullback-Leibler divergence.

$$\mathcal{L}_{\text{ST}} = \gamma^2 \cdot D_{\text{KL}}\left(\text{log\_Softmax}\left(\frac{out_s}{\gamma}\right), \text{Softmax}\left(\frac{out_t}{\gamma}\right)\right) \tag{3}$$

We hypothesize equation $\mathcal{L}_{\text{overall}}$ (equation 1) to be a powerful approach to achieve better performance and generalization in the IMU2CLIP model for the following reasons:

- ***Complementarity of Losses:*** Contrastive loss and Distillation loss have different focuses, and they complement each other. InfoNCE loss is designed to maximize the mutual information between different views of the same data instance. It helps to learn robust and invariant features by forcing the model to maximize the similarity of positive pairs and minimize the similarity of negative pairs. On the other hand, Distillation loss transfers knowledge learned by large pre-trained models (CLIP) to the smaller student model (IMU2CLIP). The teacher model has already learned valuable representations and can guide the student model to good solution space.
- ***Regularization Effect:*** Combining these two losses can act as a form of regularization. Each loss enforces different properties on the learned representation, and using them together can help avoid overfitting to a single objective.
- ***Leveraging Pretrained Models:*** The distillation loss, in particular, leverages the fact that large language models like CLIP have been trained on vast amounts of data and have learned to extract useful features from text and images. This loss encourages the student model to learn similar feature representations, which can significantly improve its performance even when trained on smaller datasets.

We further optimize the multi-objective loss by exploring different variants of distillation loss.

Similarity Preservation loss (22) in equation 4 focuses on maintaining the similarity relationships between data points in the student model's embedding space. We hypothesize that similarity preservation can improve generalization while transferring knowledge from our teacher to student model while preserving the structural information of the teacher model's output space.

$$\mathcal{L}_{\text{SP}} = \frac{1}{B}\sum_{i=1}^{B}\left(\frac{G_s}{||G_s||_2} - \frac{G_t}{||G_t||_2}\right)^2. \tag{4}$$

where $G_t = fm_t fm_t^T$ and $G_s = fm_s fm_s^T$ are Gram matrices in student and teacher model.

Neuron selectivity Transfer Loss (11) in equation 5 focuses on transferring the selectivity of individual neurons from the teacher model to the student model. It refers to the extent to which a neuron in a deep neural network responds to specific features or patterns in the input data. Highly selective neurons can be more informative and contribute to the model's overall performance. We hypothesize that neuron selectivity transfer enhances feature extraction capabilities and provides compact, efficient representation to a student model by using fewer neurons for same information

$$\mathcal{L}_{\text{NST}} = \mathrm{E}_{s,t} \cdot K(f(s), f(s)) \\ +K(f(t), f(t)) - 2K(f(s), f(t)) \tag{5}$$

where $E_{s,t}$ denotes the expectation over all student and teacher feature maps, $K(f(s), f(t))$ denotes the polynomial kernel between the student's normalized feature map $f(s)$ and the teacher's normalized feature map $f(t)$, which is calculated as square of the sum of their product.

Attention Transfer Loss (25) in equation 6 focuses on transferring the attention maps from the teacher model to the student model. An attention map refers to a heatmap that shows which parts of the input the model is "paying attention to" or focusing on to make its predictions. Attention maps provide a way to interpret what the model is doing, as they highlight the areas of the input that are most relevant to the model's prediction. We hypothesize that attention transfer enhances feature focusing capabilities and help improve model performance on tasks that require feature interpretabilities.

$$\mathcal{L}_{\text{AT}} = \text{MSE}\left(\frac{\sum |f(s)|^p}{\| \sum |f(s)|^p \|_2}, \frac{\sum |f(t)|^p}{\| \sum |f(t)|^p \|_2}\right) \tag{6}$$

where: $f(s)$ and $f(t)$ are normalized feature maps of student and teacher, and p is a hyperparameter that determines the power to which the absolute value of the feature map is raised.

Correlation Congruence (20) in equation 7 focuses on preserving the complex correlation structure of the teacher model's feature representations in the student model. We hypothesize that correlation congruence help improve model performance on tasks that require understanding of feature correlations.

$$\mathcal{L}_{\text{CC}} = \text{MSE}\Bigg(\sum_{p=0}^{P} \frac{e^{-2\gamma}(2\gamma)^p}{p!}(f(s)f(s)^T)^p, \\ \sum_{p=0}^{P} \frac{e^{-2\gamma}(2\gamma)^p}{p!}(f(t)f(t)^T)^p\Bigg) \tag{7}$$

where: $f(s)$ and $f(t)$ are the normalized feature maps of student and teacher, $e^{-2\gamma}(2\gamma)^p/p!$ is the scaling factor for each term in the series expansion and $(f(s)f(s)^T)^p$ and $(f(t)f(t)^T)^p$ are the pth powers of the similarity matrices of the student and the teacher.

Probabilistic Knowledge Transfer (19) in equation 8 allows the student to learn from teacher model's uncertainty which is particularly useful when data is noisy or contains outliers. We hypothesize that models trained with probabilistic knowledge transfer are better calibrated (meaning their predicted probabilities accurately reflect the true likelihood of each outcome), robust and thus, enhance model performance.

$$\mathcal{L}_{\text{PKT}} = \mathrm{E}\left[p(t) \log\left(\frac{p(t) + \epsilon}{p(s) + \epsilon}\right)\right] \tag{8}$$

where: $p(t)$ and $p(s)$ are the conditional probabilities of the teacher and the student. These probabilities are calculated from cosine similarities, which are first normalized to the range [0,1] and then normalized to sum to 1 across each row. E[.] denotes the expectation operation, which in this case amounts to taking the mean over all elements. ϵ is a small constant added for numerical stability, to prevent division by zero or taking the logarithm of zero.

Relational KD (18) in equation 9 focuses on maintaining the relational information between data points in student model's embedding space. We hypothesize that Relational KD can improve model generalization and robustness while transferring knowledge from our teacher to student model and preserving the complex relational structural information of the teacher model's output space.

$$\mathcal{L}_{\text{RKD}} = w_{\text{dist}} \cdot \text{Smooth}\left(D_{\text{s}}, D_{\text{t}}\right) \\ +w_{\text{angle}} \cdot \text{Smooth}\left(A_{\text{s}}, A_{\text{t}}\right) \tag{9}$$

where: w_{dist} and w_{angle} are weights for the distance-based and angle-based loss parts, D_{s} and D_{t} are normalized pairwise distances among the student's and teacher's features , A_{s} and A_{t} are the angle proxies - cosine similarities of pairwise feature differences for student and teacher and $\text{Smooth}(., .)$ denotes Smooth L1 loss, which is less sensitive to outliers than Mean Squared Error loss.

Hint-based Training ("FitNets") (21) in equation 10 focuses on using intermediate representations (hints) from teacher model to guide training of student model. FitNets loss is a measure of the discrepancy between the intermediate representations of the teacher and student models. We hypothesize that FitNets provide guidance effectively and efficiently on every intermediate.

$$\mathcal{L}_{\text{FitNets}} = \text{MSE}(f_{\text{s}}, f_{\text{t}}) \tag{10}$$

where f_{s} and f_{t} represent the student and teacher feature maps.

Activation Boundaries (9) in equation 11 focuses on preserving the decision making process of the teacher model in the student model. We hypothesize that by focusing on such preservation, activation boundaries make the student model more robust to variations in the input data and improve its ability to generalize to unseen data.

$$\mathcal{L}_{\text{AB}} = \text{mean}\left((f_{\text{s}} + m)^2 \cdot I(f_{\text{s}} > -m \wedge f_{\text{t}} \leq 0) \\ +(f_{\text{s}} - m)^2 \cdot I(f_{\text{s}} \leq m \wedge f_{\text{t}} > 0)\right) \tag{11}$$

where m is the margin, f_s and f_t represent the student and teacher feature maps before activation and I is the indicator function which is 1 when the condition inside the parentheses is true and 0 otherwise.

Logits (2) in equation 12 matches the outputs of the final layer of the teacher model, before the softmax activation function with the student model. We hypothesize that Logits transfers rich information and thus improve model performance and robustness on tasks that require nuance understanding of input data.

$$\mathcal{L}_{\text{Logits}} = \text{MSE}(out_s, out_t) \tag{12}$$

where out_s and out_t are the outputs of student and teacher models.

3.3 Energy based Loss

We explore a variation of distance-based loss called Triplet Loss, an energy based loss, used to learn an embedding space of an input image, where similar inputs are mapped to nearby points and dissimilar inputs are mapped to distant points (24). It has three major components namely the anchor, positive and negative samples. Triplet loss aims at increasing the distance between the anchor and the negative examples, while decreasing the distance between the anchor and positive samples.

The idea of triplet loss in the context of contrastive learning is to use triplets of data points in the loss function, where each triplet consists of an anchor point, a positive point, and a negative point. The anchor and positive points are chosen from the same class or category, while the negative point is chosen from a different class or category. The goal here is to encourage the network to learn representations such that the distance between the anchor and positive points is smaller than the distance between the anchor and negative points by a margin value. This can be expressed mathematically in equation 13.

$$\mathcal{L}_{\text{triplet}} = \sum_{i=1}^{N} \max\Big(0, \|f(x_i^a) - f(x_i^p)\|^2 - \|f(x_i^a) - f(x_i^n)\|^2 + \text{margin}\Big) \tag{13}$$

where f is the neural network model, x_a is the anchor point, x_p is the positive point, x_n is the negative point, $||.||$ denotes the L2 norm, which is the Euclidean distance and margin is a hyperparameter that controls the minimum desired difference between the positive and negative distances.

4 Experiments

4.1 Datasets

We utilize the Ego4D dataset (7) for all the experimentation and model developments. Ego4D is a publicly available egocentric dataset that consists of 3,670 hours of video collected by 923 unique participants from 74 worldwide locations in 9 different countries. Egocentric data is a subcategory of visual data that refers to the images and videos captured in a first-person perspective by a wearable camera. *See Table 1* depicts the original train/val/test split. The data used for our experiments is greater than 1 TB (includes video, IMU, and text data) and we have successfully preprocessed and downscaled the quality (video data primarily) so that it's quicker to run the experiments. Even though the original dataset is around 5TB in size, we shall only be using 1.25TB, as all the videos do not have IMU data attributed to them and hence won't be useful in our case.

Ego4d	Tra.	Val.	Tst.
# of Media files	1444	161	688
Total Media Durations	540h	60h	265h
# of IMU↔Text/Video windows (5s)	528K	68K	266K
# of IMU→4 classes windows (5s)	1552	760	241

Table 1: Dataset Statistics for Ego4D.

4.2 Baseline: Reproduction and Enhancement

We compare our proposed enhancements with the MW2 architecture (baseline) which is a stacked RNN architecture proposed in the IMU2CLIP (16). Please refer to figure 12 for the IMU encoder architecture.

a) Batch Size : After successful data preprocessing, we reciprocated the results of IMU → Text module. Table 3 depicts the comparison between the actual and reciprocated baseline results. We have also found that increasing batch size had an improvement in the results. This can be attributed to the fact that in self-supervised learning approaches involving contrastive learning, using a larger batch size can provide several benefits (3). With a larger batch size, the model can learn more meaningful patterns and relationships by accessing a more representative sample of the overall data distribution. Moreover, larger batch sizes can reduce the variance in gradient estimates, leading to more stable training and better model generalization. In our experiments, we varied the training batch size from the original 32 to 256 to investigate its impact on the model's performance. Specifically, we found that a batch size of 256 gave us good results, as it generated 700% more negative examples compared to old batch size, thereby enhancing the model's ability to discriminate between similar and dissimilar examples.

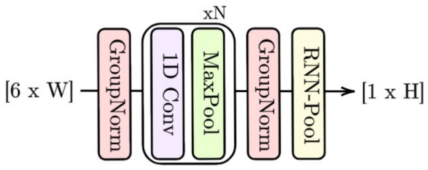

Figure 3: Stacked RNN architecture for the IMU encoder used in baseline IMU2CLIP Model

b) Normalization Techniques: Normalization techniques are important in self-supervised machine learning for several reasons: *Enhancing learning stability*:

Normalization helps stabilize the learning process by reducing the range of input values. Self-supervised learning often involves tasks such as predicting missing parts of an image or inferring relationships between different data samples. By normalizing the input data, we ensure that the learning algorithm focuses on the patterns and relationships rather than being affected by the scale or magnitude of the input features.

Improving convergence speed: Normalization can accelerate the convergence of the learning algorithm. When training deep neural networks in self-supervised learning, normalization techniques like batch normalization or layer normalization can help alleviate the vanishing or exploding gradients problem. This enables faster and more stable convergence during training.

Facilitating generalization: Normalization helps the model generalize well to unseen data. By normalizing the input features, we make them more comparable and reduce the influence of irrelevant variations in the data. This enables the model to capture more meaningful and discriminative patterns, leading to better generalization performance.

Preserving relative relationships: In self-supervised learning, the model learns from the inherent relationships between data samples. Normalization techniques preserve these relationships by maintaining the relative differences between the features while removing any absolute scale or bias. This allows the model to focus on learning the meaningful relationships and dependencies between the data samples.

Hence, application of various normalization techniques, especially in a self-supervised learning setting carries significant prominence. We have trained the model using three different normalization techniques, Instance Normalization (IN), Layer Normalization (LN) and the base line's default, Group Normalization (GN). The results are shown in the following graphs, and the performances accross various recall metrics is reported in Table 2. Evidently, Layer Normalization works best over GN and IN, and let us analyze the intuition behind. It can be attributed to the following reasons :

Independence of batch size: Layer Normalization operates independently of batch size, making it suitable for situations where the batch size is small or even when using a single sample for training. In self-supervised learning, where the batch size can be small due to computational constraints, Layer Normalization's ability to normalize effectively with smaller batches can be advantageous.

Handling spatial correlations: Self-supervised tasks often involve data with spatial correlations, such as images or sequences (of IMU signals in this setting). Layer Normalization, by normalizing within each layer independently, effectively captures the spatial correlations within the features, which can be beneficial for modeling complex patterns and relationships.

Reduced sensitivity to group size: Group Normalization divides the channels into groups and normalizes each group separately. In self-supervised tasks, the choice of group size can impact the performance of Group Normalization. Layer Normalization does not rely on grouping channels, making it less sensitive to the choice of group size and more flexible in adapting to different network architectures.

Better performance on small batch sizes: Self-supervised learning often involves training with small batch sizes, as mentioned earlier. Layer Normalization has shown to perform better than Group Normalization in scenarios with small batch sizes, where Group Normalization may struggle due to insufficient statistics within each group.

Instance Normalization: Recent studies have shown that normalization techniques such as Batch Normalization and Group Normalization may not work well in contrastive learning tasks. This is because Batch Normalization and Group Normalization normalize the feature statistics across the batch or group dimension, which can cause disagreement between the augmented views. Instead, normalization techniques that normalize the feature statistics across the channel dimension, such as Instance Normalization, is preferred in contrastive learning tasks. Specifically, recent studies have shown that using Instance Normalization can significantly improve the performance of contrastive learning models. Instance normalization normalizes the features across the channel dimension, which can reduce the disagreement between the augmented views and improve the quality of learned representations. In addition, it is less sensitive to the batch size, which can improve the stability and performance of contrastive learning models. In (23) by Kwonjoon Lee et al., Instance Normalization is used over Batch Normalization to reduce the disagreement between the augmented views in the "align-and-unify" approach. On the contrary, we found that Group normalization is better suited here over Instance Normalization, inferred from the results.

c) Knowledge Distillation Losses: For distillation losses, we optimize the parameters with Adagrad with batch size as 256, learning rate as $2e-4$, epsilon as 10^{-8}, and decay as 0.1. See table 3

d) IMU Summarizer: Towards proposing a new downstream task to evaluate the practical effectiveness of the IMU encoder, we have designed an IMU summarizer that takes in a sensor's IMU recordings as input, encodes them into IMU aligned CLIP embeddings and outputs the summary of the activity associated with the IMU readings. For the duration of our project, we have added an intermediary task after the encoding stage, where we predicted the activity associated with each frame corresponding to their timestamp. This activity prediction is done in a supervised manner with each frame having soft annotations of their activity associated in the Ego4D dataset (7). Later, we fed the timestamp-activity data to large language model interfaces like ChatGPT, and Bard to output the summary of the IMU frame. In doing so, we realized that this pipeline might

	IMU → Text				**Text → IMU**			
Config	R@1	R@10	R@50	MRR	R@1	R@10	R@50	MRR
Group Norm(16)	4.86	18.75	48.26	0.104	4.17	15.62	43.06	0.084
Instance Norm	3.81	20.48	50	0.095	3.81	20.48	50	0.080
Layer Norm	5.55	27.43	58.68	0.125	5.20	28.47	62.84	0.130

Table 2: Normalization Evaluations

	IMU → Text				**Text → IMU**			
Configurations	R@1	R@10	R@50	MRR	R@1	R@10	R@50	MRR
Baseline(16) (BS 32)	5.21	25.00	60.42	0.123	7.29	28.82	60.07	0.143
Reproduced (BS 32)	4.51	22.56	55.20	0.106	4.17	25.00	56.94	0.110
w/ Baseline + Similarity Preservation	4.86	23.61	58.33	0.120	5.21	27.77	56.94	0.122
w/ Baseline + Neuron Transfer	4.86	**26.38**	57.29	0.119	3.81	25.69	58.68	0.110
w/ Baseline + Attention Transfer	4.86	25	60.06	0.124	**7.29**	**30.9**	59.03	**0.143**
w/ Baseline+ "Focused" Attention Transfer	**5.21**	**26.39**	**60.76**	**0.125**	5.9	26.74	**59.72**	0.131
w/ Baseline+ Correlation Congruence	**5.55**	**26.04**	59.37	0.108	3.47	26.04	57.98	0.104
w/ Baseline + Probabilistic Knowledge Transfer	4.86	23.61	58.33	0.120	4.86	26.38	59.72	0.118
w/ Baseline + Relational KD	2.08	13.88	40.97	0.071	3.81	14.23	40.62	0.083
w/ Baseline + FitNets	4.16	22.22	60.06	0.107	5.20	25.34	59.03	0.122
w/ Baseline + Activation Boundries	3.28	**27.08**	57.98	0.117	4.16	27.77	59.37	0.120
w/ Baseline+ Soft Target	**5.21**	**25.34**	**61.11**	0.123	4.51	26.04	57.98	0.119
w/ Baseline+ Logits	**6.25**	25	59.37	0.129	3.12	28.47	57.98	0.111
w/ Triplet CS (BS 32)	1.04	6.597	25	0.0369	1.388	6.25	24.65	0.038
w/ Triplet (BS 32)	0.694	5.902	21.87	0.0308	0.694	5.55	25.34	0.032

Table 3: Text↔IMU retrieval performances of the pre-trained models on Ego4D, with different modalities used for training. Here, CS stands for cyclic shift, and BS stands for batch size which is 256 when not mentioned.

not be the most ideal way of summarising activity since the summaries weren't very descriptive and have designed a new pipeline. In the modified pipeline, we generate the IMU-aligned CLIP embeddings for our entire dataset and create a new (embedding, text narration) dataset which we aim to finetune on larger models like T5, Bert so as to generate summary directly from the IMU embeddings itself. This would be the future direction of our work.

4.3 Evaluation Metrics

We evaluate our proposed model with mean reciprocal rank and recall at different k values of 1, 10 and 50 for IMU signal window retrieval given a textual query or a video frame. By examining recall at different k values we gain better insights at our model's performance and identify areas where performance is specifically effective or ineffective. However, all k values are equally weighted for evaluation which may not be the case while looking at higher k values. Hence, we also look at the MRR. A higher MRR indicates better model performance, as it means that relevant items are retrieved at higher ranks on average.

4.4 Results

Distilling knowledge in cross-modal contrastive learning where the teacher model is CLIP with text descriptions, and student model being IMU2CLIP with IMU embeddings is a worthy experiment. While performing text↔IMU retrieval tasks we observe that a combination of contrastive and distillation losses perform much better than the baseline. Some noteworthy observations include Baseline + Attention transfer (p =2) which shows an improvement of 7.21% individually on recall@k = 10 and reaches the baseline benchmark in most metrics. Other special mentions are Baseline + Logits which perform significantly better in IMU→text retrieval and Baseline + Soft Target which perform marginally better in IMU→text retrieval. We also explored to increase the focus of attention transfer (p =5) and observed marginally better performance in IMU→text retrieval. Reason for not achieving significant better performance is due to too much removal of features ("too focused").

Distillation losses failed to optimize the model in a single objective loss setting while trying to capture the weights for the network despite using a stabilised version (Mean of Individual distillation loss). This is most likely due to the model not becoming robust and invariant to features which is the primary function of contrastive losses. The application of Attention Transfer combined with InfoNCE loss outperformed other combinations of distillation losses with InfoNCE due to the following reasons:

- An Attention mechanism allows for more robust and richer representation of complex temporal patterns and relationships present in the IMU embeddings. Moreover it has the capability to transfer such dependencies from teacher to student model.
- InfoNCE loss, when coupled with Attention Trans-

fer, can further amplify the benefits of noise reduction. InfoNCE inherently reduces noise by emphasizing positive pairs over negative pairs in the representation space. When combined with the attention mechanism's ability to suppress irrelevant features, this results in superior noise reduction and hence, better performance.

Overall InfoNCE and Attention Transfer are complementary to each other. InfoNCE focuses on pushing apart dissimilar examples and pulling together similar ones, while Attention Transfer focuses on ensuring the student model pays attention to the same parts of the input as the teacher. Thus, together they provide a comprehensive and balanced learning objective for the student model.

We have also found that tuning different parameters like batch size and normalization methods seemed to significantly improve the performance of the encoder. We are still early in finding the optimal hyperparameters and we are hopeful in this approach as well.

Best Model : The best model so far through our experimentations has been the one combining the best-performing knowledge distiller in cross-modal contrastive learning with the IMU encoder working on Layer Normalization to derive our ***final architecture framework*** and compare the performance with the baseline benchmark. We have observed an average increase in the Mean Reciprocal Rank from the baseline by 6.2% and in the Recall performance by 9.6%.

4.5 Dimensionality Reduction and Visualization

In order to visually analyze the structure of the high-dimensional embeddings generated by our IMU encoder trained with different loss functions, we utilized t-Distributed Stochastic Neighbor Embedding (t-SNE)(15).For each set of embeddings generated by models trained with different loss functions, we applied the t-SNE technique, reducing the dimensionality from the original dimensionality (1024) to a 2-dimensional space that can be easily visualized.Upon plotting the 2D t-SNE embeddings, several patterns emerged, allowing us to draw comparisons across the different models. The visualizations can be found below.

For the model trained with Attention Transfer, we observed that the embeddings tended to form distinct clusters, corresponding closely with our activities. This suggests that the model is effectively learning to differentiate between the different classes and is potentially a strong choice for our alignment task.

Conversely, for the model trained with Focussed Attention Transfer, the embeddings appear more scattered, indicating a potential lack of distinct activity separation. This result implies that Focussed attention transfer might not be as effective in learning the nuances of our specific alignment task, resulting in embeddings that are less distinct and more homogenous.

The model trained with the baseline loss of InfoNCE produced results somewhat intermediate between the previous two. While it did exhibit some degree of class separation, the distinction was not as clear as with Attention Transfer. Further investigation would be necessary to understand the nuances of its performance.

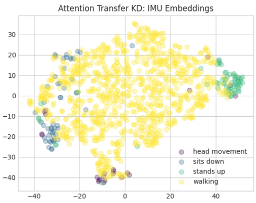

Figure 4: t-SNE Visualisation of CLIP aligned IMU Embeddings for Attention Transfer (p = 2)

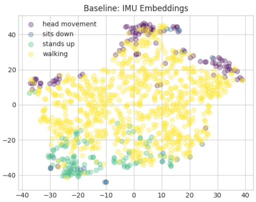

Figure 5: t-SNE Visualisation of CLIP aligned IMU Embeddings for Baseline

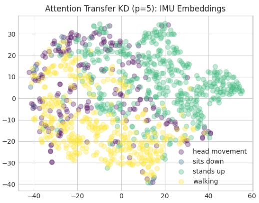

Figure 6: t-SNE Visualisation of CLIP aligned IMU Embeddings for "Focused" Attention Transfer (p = 5)

4.6 Future Scope

Our next set of work would focus more on the downstream applications of the IMU2CLIP model. Some of which include the IMU Summarizer, and Hand Object Segmentation. Aligning multiple modalities in a single latent space is a very promising and emerging research area right now and we hope that IMU2CLIP serves as a good catalyst in this process.

5 Acknowledgements

We would like to express our sincere gratitude to our mentors - Dr. Shane Moon and Aparajita Saraf from Meta Reality Labs for their constant support and valuable inputs and the entire teaching staff for their crucial feedbacks at every stage. We also like to thank our Ph.D mentor Shiv Shankar for his support and feedback.

References

[1] Alec Radford, Jong Wook Kim, C.H.A.R.G.G.S.A.G.S.A.A.P.M.J.C.G.K.I.S.: Learning transferable visual models from natural language supervision. https://arxiv.org/pdf/2103.00020.pdf (2021)

[2] Ba, L.J., Caruana, R.: Do deep nets really need to be deep? (2014)

[3] Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations (2020)

[4] Chen, X., Zhang, K., Liu, H., Leng, Y., Fu, C.: A probability distribution model-based approach for foot placement prediction in the early swing phase with a wearable imu sensor. IEEE Transactions on Neural Systems and Rehabilitation Engineering **29**, 2595–2604 (2021). https://doi.org/10.1109/TNSRE.2021.3133656

[5] Damen, D., Doughty, H., Farinella, G.M., Fidler, S., Furnari, A., Kazakos, E., Moltisanti, D., Munro, J., Perrett, T., Price, W., Wray, M.: The epic-kitchens dataset: Collection, challenges and baselines. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) **43**(11), 4125–4141 (2021). https://doi.org/10.1109/TPAMI.2020.2991965

[6] Gou, J., Yu, B., Maybank, S.J., Tao, D.: Knowledge distillation: A survey. International Journal of Computer Vision **129**(6), 1789–1819 (mar 2021). https://doi.org/10.1007/s11263-021-01453-z, https://doi.org/10.1007%2Fs11263-021-01453-z

[7] Grauman, K., Westbury, A., Byrne, E., Chavis, Z., Furnari, A., Girdhar, R., Hamburger, J., Jiang, H., Liu, M., Liu, X., Martin, M., Nagarajan, T., Radosavovic, I., Ramakrishnan, S.K., Ryan, F., Sharma, J., Wray, M., Xu, M., Xu, E.Z., Zhao, C., Bansal, S., Batra, D., Cartillier, V., Crane, S., Do, T., Doulaty, M., Erapalli, A., Feichtenhofer, C., Fragomeni, A., Fu, Q., Gebreselasie, A., Gonzalez, C., Hillis, J., Huang, X., Huang, Y., Jia, W., Khoo, W., Kolar, J., Kottur, S., Kumar, A., Landini, F., Li, C., Li, Y., Li, Z., Mangalam, K., Modhugu, R., Munro, J., Murrell, T., Nishiyasu, T., Price, W., Puentes, P.R., Ramazanova, M., Sari, L., Somasundaram, K., Southerland, A., Sugano, Y., Tao, R., Vo, M., Wang, Y., Wu, X., Yagi, T., Zhao, Z., Zhu, Y., Arbelaez, P., Crandall, D., Damen, D., Farinella, G.M., Fuegen, C., Ghanem, B., Ithapu, V.K., Jawahar, C.V., Joo, H., Kitani, K., Li, H., Newcombe, R., Oliva, A., Park, H.S., Rehg, J.M., Sato, Y., Shi, J., Shou, M.Z., Torralba, A., Torresani, L., Yan, M., Malik, J.: Ego4d: Around the world in 3,000 hours of egocentric video (2021). https://doi.org/10.48550/ARXIV.2110.07058, https://arxiv.org/abs/2110.07058

[8] He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning (2020)

[9] Heo, B., Lee, M., Yun, S., Choi, J.Y.: Knowledge transfer via distillation of activation boundaries formed by hidden neurons (2018)

[10] Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network (2015)

[11] Huang, Z., Wang, N.: Like what you like: Knowledge distill via neuron selectivity transfer (2017)

[12] Jiang, Y., Song, L., Zhang, J., Song, Y., Yan, M.: Multi-category gesture recognition modeling based on semg and imu signals. Sensors **22**(15) (2022). https://doi.org/10.3390/s22155855, https://www.mdpi.com/1424-8220/22/15/5855

[13] Karakas, S., Moulon, P., Zhang, W., Yang, N., Straub, J., Ma, L., Lv, Z., Argall, E., Berenger, G., Schmidt, T., Somasundaram, K., Baiyya, V., Bouttefroy, P., Sawaya, G., Lou, Y., Huang, E., Shen, T., Caruso, D., Souti, B., Sweeney, C., Meissner, J., Miller, E., Newcombe, R.: Aria data tools. https://github.com/facebookresearch/aria_data_tools (2022)

[14] Koch, G., Zemel, R., Salakhutdinov, R., et al.: Siamese neural networks for one-shot image recognition. In: ICML deep learning workshop. vol. 2. Lille (2015)

[15] van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of Machine Learning Research **9**(86), 2579–2605 (2008), http://jmlr.org/papers/v9/vandermaaten08a.html

[16] Moon, S., Madotto, A., Lin, Z., Dirafzoon, A., Saraf, A., Bearman, A., Damavandi, B.: Imu2clip: Multimodal contrastive learning for imu motion sensors from egocentric videos and text. arXiv preprint arXiv:2210.14395 (2022)

[17] van den Oord, A., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. CoRR **abs/1807.03748** (2018), http://arxiv.org/abs/1807.03748

[18] Park, W., Kim, D., Lu, Y., Cho, M.: Relational knowledge distillation (2019)

[19] Passalis, N., Tefas, A.: Learning deep representations with probabilistic knowledge transfer (2019)

[20] Peng, B., Jin, X., Liu, J., Zhou, S., Wu, Y., Liu, Y., Li, D., Zhang, Z.: Correlation congruence for knowledge distillation (2019)

[21] Romero, A., Ballas, N., Kahou, S.E., Chassang, A., Gatta, C., Bengio, Y.: Fitnets: Hints for thin deep nets (2015)

[22] Tung, F., Mori, G.: Similarity-preserving knowledge distillation (2019)

[23] Wang, T., Isola, P.: Understanding contrastive representation learning through alignment and uniformity on the hypersphere (2022)

[24] Weinberger, K.Q., Blitzer, J., Saul, L.: Distance metric learning for large margin nearest neighbor classification. In: Weiss, Y., Schölkopf, B., Platt, J. (eds.) Advances in Neural Information Processing Systems. vol. 18. MIT Press (2005), https://proceedings.neurips.cc/paper/2005/file/a7f592cef8b130a6967a90617db5681b-Paper.pdf

[25] Zagoruyko, S., Komodakis, N.: Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer (2017)

APPENDIX

A Retrieval Results Plot for Distilling Knowledge in Cross-Modal Contrastive Learning

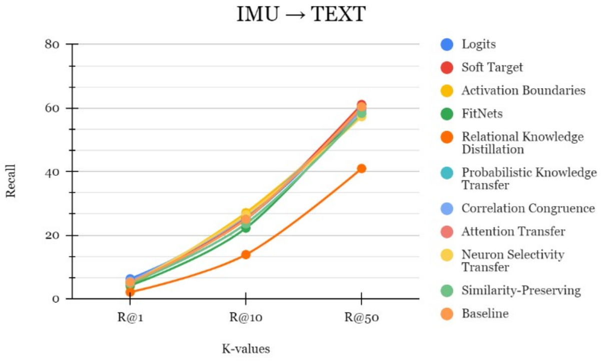

Figure 7: Recall results for IMU→text retrieval

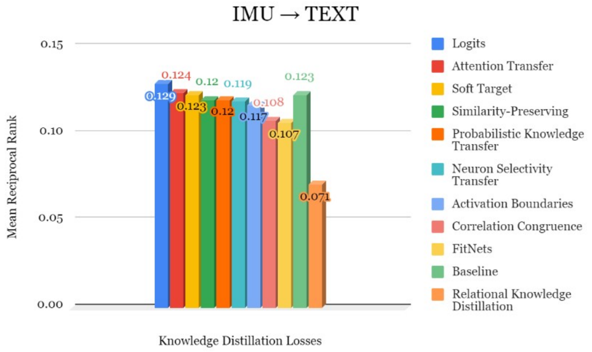

Figure 8: Mean Rank Reciprocal results for IMU→text retrieval

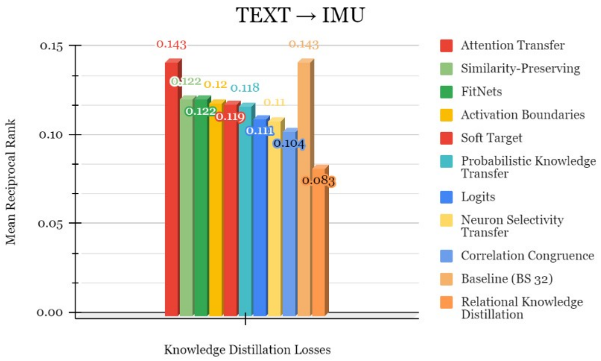

Figure 9: Recall results for text→IMU retrieval

B Retrieval Results Plot for Normalizations Performance

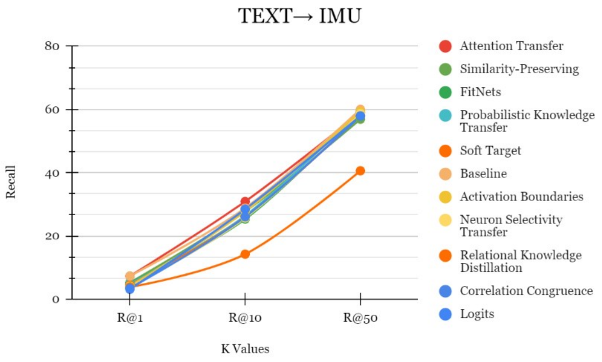

Figure 10: Mean Rank Reciprocal results for text→IMU retrieval

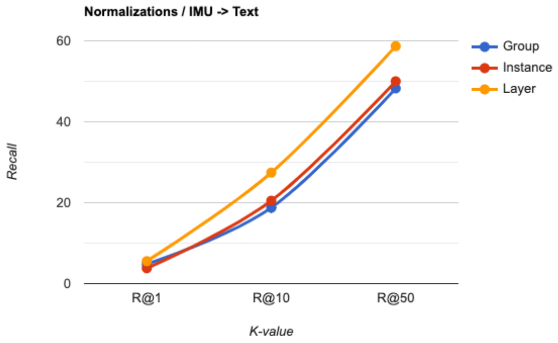

Figure 11: Normalizations Performance in IMU to Text

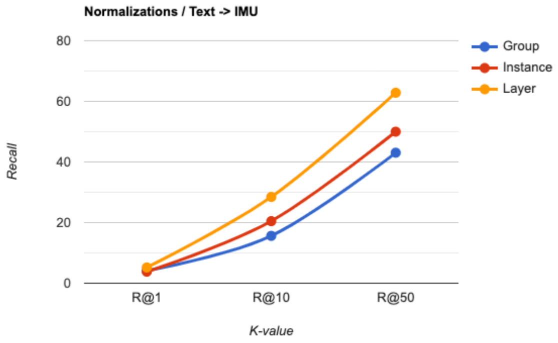

Figure 12: Normalizations Performance in Text to IMU