## A. Radar Chart Figure 1 Details

We first describe how we plot the radar chart in Figure 1. Each axis denotes a specific metric on one video understanding task. Each vertex denotes a ratio relative to our performance, which is computed by normalizing the performance of either LaViLa or previous SOTA by that of LaViLa, and is in the range of $(0, 1]$. For illustrative purpose, we set the radar chart's origin to be 80% and outermost frame to be 100% so that the interval between neighboring lattices to be 5%. The numbers annotated next to the vertices are *absolute value* of performance *without normalization*. Note that in other radar charts [69, 80], the axes have different scales and interval values while the origin is not valid, which may lead to potential fallacies.

Next we elaborate the evaluation metrics and previous state-of-the-arts in each axis. For EK-100 MIR and CharadesEgo, we compare our method to EgoVLP [39] in the fine-tuned settings. For EgoMCQ, we compare our method to EgoVLP [39] in the zero-shot settings. For EGTEA recognition, the previous state-of-the-art is MTCN [33]. For EK-100 CLS, we plot the action-level top-1 accuracy after fine-tuning as it is the primary metric proposed in [14] (and used in the EPIC challenges). The previous state-of-the-art is Multiview Transformer (MTV) [79] pre-trained on a private dataset. For UCF-101 and HMDB-51 classification, we report the linear-probing mean accuracy following MIL-NCE [44]. The previous state-of-the-art is TAN [25].

## B. LaViLa Details

The algorithm of training LaViLa is given in Algorithm 1. The loss is based on the CLIP [49]'s symmetric cross-entropy loss over the similarity scores of samples in the batch $\widetilde{\mathcal{B}}_l \cup \widetilde{\mathcal{B}}_u$ with minimal modifications. We apply two separate temperatures $(\tau_r, \tau_n)$ for embeddings from rephrased pairs and pseudo-captioned ones respectively,

$$\mathcal{L} = -\frac{1}{2N}\sum_{i=1}^{N}\left(\log\frac{\exp(\frac{\mathbf{v}_i^\top \mathbf{u}_i}{\tau_i})}{\sum_{j=1}^{N}\exp(\frac{\mathbf{v}_i^\top \mathbf{u}_j}{\sqrt{\tau_i\tau_j}})} + \log\frac{\exp(\frac{\mathbf{u}_i^\top \mathbf{v}_i}{\tau_i})}{\sum_{j=1}^{N}\exp(\frac{\mathbf{u}_i^\top \mathbf{v}_j}{\sqrt{\tau_i\tau_j}})}\right). \quad (4)$$

We ablate different choices of temperatures in Table 12.

**Algorithm 1** One step of training LaViLa

**Require:** A subset of narrated (unnarrated) clips $\mathcal{B}_l$ ($\mathcal{B}_u$)
  clips with LM-generated narrations $\widetilde{\mathcal{B}}_l = \{\}, \widetilde{\mathcal{B}}_u = \{\}$
  **for** $(x_i, y_i) \in \mathcal{B}_l$ **do**
    $u \sim U(0, 1)$ ▷ Uniform sample between 0 and 1
    **if** $u < 0.5$ **then** ▷ Query Rephraser
      $y'_i \sim p_{\text{Rephraser}}(y'|y_i), \tau_i \leftarrow \tau_r$
    **else** ▷ Query Narrator
      $y'_i \sim p_{\text{Narrator}}(y'|x_i), \tau_i \leftarrow \tau_n$
    **end if**
    $\widetilde{\mathcal{B}}_l \leftarrow \widetilde{\mathcal{B}}_l \cup \{(x_i, y'_i, \tau_i)\}$
  **end for**
  **for** $x_i \in \mathcal{B}_u$ **do**
    $y'_j \sim p_{\text{Narrator}}(y'|x_i), \tau_j \leftarrow \tau_n$
    $\widetilde{\mathcal{B}}_u \leftarrow \widetilde{\mathcal{B}}_u \cup \{(x_j, y'_j, \tau_j)\}$
  **end for**
  Train $\mathcal{F}_{\text{LaViLa}}(x, y)$ with the batch $\widetilde{\mathcal{B}}_l \cup \widetilde{\mathcal{B}}_u$ using Eq 4.

## C. Dataset Details

In this section, we provide details of the datasets where we conduct experiments.

**Ego4D**. Ego4D contains 3,670 hours of egocentric videos with temporally dense narrations. Each narration has a timestamp and an associated free-form sentence. We construct the video-text clip pairs that are used for pre-training following [39]. First, we exclude 2,429 videos that appear in the validation and test sets of the Ego4D benchmark. Next, we determine the each clip's interval using the contextual variable-length clip pairing strategy in [39]. Finally, we drop the narrations that either contain "#unsure"/"#Unsure" tags or are shorter than 4 words. This results in 4,012,853 video-text clip pairs with an average clip length of $1(\pm0.9)$ second. For the excluded videos, we also pre-process similarly and obtain 1,260,434 video-text clip pairs. We only use them as validation split to measure the generation quality of Narrator in Table 7a.

**EK-100**. The Epic-Kitchens-100 (EK-100) dataset contains 100 hours of egocentric cooking videos. The training split has 67,217 video clips; the validation split has 9,668 video clips; the testing split has 13,092 video clips. Each clip is annotated with (1) a start and end timestamp, (2) a short textual narration, and (3) a verb and noun class that the narration belongs to. The action class can also be uniquely determined by combining the verb and the noun. In the zero-shot setting, we evaluate the pre-trained model on the validation split directly without any tuning; In the finetuned setting, we take the pre-trained model and perform end-to-end finetuning on the training split and evaluate on the validation split. For EK-100 MIR we use the textual narration while for EK-100 CLS we use the class of verb, noun, and action as the label. For EK-100 MIR, the evluation metrics are mean Average Precision (mAP) and normalized Discounted Cumulative Gain (nDCG). For EK-100 CLS, the evaluation metrics are top-1 accuracies for verb, noun, and action. Action-level accuracy is the most important one among all.

**EGTEA**. EGTEA contains 28 hours of egocentric cooking videos with gazing tracking. In our experiments, we take as input the visual frames only. The action annotations include 10,321 instances of fine-grained actions from 106 classes, with an average duration of 3.2 seconds. In the zero-shot setting, we evaluate the pre-trained model on the test set of all three splits without any tuning and report results as the mean accuracy averaged across all classes across all three splits, as Li *et al*. [37] suggested. In the finetuned setting, we follow prior works [33] and report top-1 accuracy and mean class accuracy using the first train/test split, which has 8,299/2,022 instances respectively.

**CharadesEgo.** The CharadesEgo dataset contains 7,860 videos of daily indoor activities from both third- and first-person views. The annotations are 68,536 instances of fine-grained actions from 157 classes. We use the first-person subset only, comprising 3,085 videos for training and 846 videos for testing. We report video-level mAP as the evaluation metric. In the zero-shot setting, we evaluate the pre-trained model on the test videos directly without any tuning; In the finetuned setting, we perform end-to-end finetuning on the trimmed action instances in the training split, which has an amount of 33,114 action instances.

# D. Implementation Details

## D.1. Pre-training on Ego4D

We pre-train on the video-narration pairs from Ego4D [24]. We train the model using AdamW with $(\beta_1, \beta_2) = (0.9, 0.999)$ and weight decay of 0.01 for 5 epochs. After the video-narrations pairs are augmented by NARRATOR and REPHRASER, we find the zero-shot performance keeps improving so the number of epochs is increased to 12. We use a fixed learning rate of 3e-5. The projection head after the dual-encoders is a linear layer with an output dimension of 256. We use PyTorch's native FP16 mixed precision training and gradient checkpoint. This allows us to afford a per-gpu batch size of 32 over 32 GPUs for TimeSformer-B and a per-gpu batch size of 16 over 64 GPUs for TimeSformer-L, resulting in a total batch size of 1,024. We abate these design choices in Appendix F.

For input, we first divide each video into 5-minute segments and scale the short side of the video to 288 pixels. This signifantly reduces storage and accelerates decoding. During training, we decode the corresponding segment that contains the selected clip. We randomly sample 4 frames between the start and end time of the clip and use standard `RandomResizedCrop (0.5, 1.0)` for data augmentation.

## D.2. Training NARRATOR on Ego4D

**Architecture.** For the video encoder, we use the one we obtain in Appendix D.1 and keep it frozen. We drop the global average pooling layer and attach an attention pooling module, which is instantiated by a standard cross-attention [66] and a Layer Normalization [3]. The attention pooling uses a fixed length of randomly initalized queries $\mathbf{q} \in \mathbb{R}^{N_\text{q} \times D_\text{t}}$ to attend visual features $\mathbf{v} \in \mathbb{R}^{(T \times H' \times W') \times D_\text{v}}$. This results in a fixed length of hidden states, $\text{AttentionPool}(\mathbf{q}, \mathbf{v}) \in \mathbb{R}^{N_\text{q} \times D_\text{t}}$, which will be later fed into the cross-attention module of the text decoder. This ensures the text decoder attends to the same number of visual features irrespective of the input visual resolution, *e.g*. 224×224 or 336×336. More concretely, $\text{AttentionPool}(\mathbf{q}, \mathbf{v})$ is computed as follows:

$$\begin{aligned} \mathbf{q}', \mathbf{v}' &= \text{LayerNorm}(\mathbf{q}), \text{LayerNorm}(\mathbf{v}), \\ \text{head}_i &= \text{softmax}\left(\frac{(\mathbf{q}'\mathbf{W}_Q^{(i)})(\mathbf{v}'\mathbf{W}_K)^\top}{\sqrt{d_0}}\right) \cdot (\mathbf{v}'\mathbf{W}_V), \\ \text{AttentionPool} &= \text{Concat}(\text{head}_1, \cdots, \text{head}_h) \cdot \mathbf{W}_O, \end{aligned}$$

where $\mathbf{W}_Q \in \mathbb{R}^{D_\text{t} \times d_0}$, $\mathbf{W}_{K/V} \in \mathbb{R}^{D_\text{v} \times d_0}$, and $\mathbf{W}_O \in \mathbb{R}^{(h \cdot d_0) \times D_\text{t}}$.

For the text decoder, we use GPT-2 XL [50] and keep it frozen. The video encoder and the text decoder is bridged by a cross-attention module. Each cross-attention module comprises a cross-attention layer followed by a feed-forward network (FFN). Layer Normalization is added at the beginning of both cross-attention and FFN. We add `tanh`-gating [27] with an initial value of zero. We insert one cross-attention module every two GPT2-Blocks in GPT2 XL to save memory. Both the attention pooling and cross-attention modules are learnable parameters, which take less than 30% of the total parameters.

We train NARRATOR on the ground-truth video-narration pairs from Ego4D [24]. The training recipe mostly follows the one for pre-training the dual-encoders except that we use FP32 to train NARRATOR because PyTorch's native FP16 mixed-precision leads to training instability. We use the video-text clip pairs from the Ego4D's validation videos to compute the word-level classification accuracy and perplexity. We select the model with the highest accuracy as well as lowest perplexity, which is often reached after 3∼4 epochs. It takes around 2 days to train a NARRATOR using 32 V100 GPUs.

## D.3. Multi-Instance Retrieval on EK-100

We fine-tune the pre-trained model on EK100 using AdamW with $(\beta_1, \beta_2) = (0.9, 0.999)$ and weight decay of 0.01. We use cosine annealing with warmup, where the base learning rate starts from 1e-6, linearly increases to a peak of 3e-3 in the first epoch and then gradually decreases to 1e-5 following a half-wave cosine schedule. We apply the multi-instance max-margin loss [75] with a margin value of 0.2. We use a per-gpu batch size of 16 over 8 GPUs for TimeSformer-B and a per-gpu batch size of 4 over 32 GPUs for TimeSformer-L. We use a stochastic depth ratio of 0.1 in the backbone.

For the input, we represent each video clip with 16 sampled frames at both training and testing time. At training time, We scale the short side of the video to 256 pixels and then take a 224×224 crop while at testing time, we scale the short side to 224 pixels and take the center 224×224 crop.

| Method (Backbone) | Pretrain | Top-1 accuracy | | |
|---|---|---|---|---|
| | | Verb | Noun | Action |
| IPL (I3D) [71] | K400 | 68.6 | 51.2 | 41.0 |
| ViViT-L [2] | IN-21k+K400 | 66.4 | 56.8 | 44.0 |
| MoViNet [34] | N/A | **72.2** | 57.3 | 47.7 |
| MTV [79] | WTS-60M | 69.9 | **63.9** | 50.5 |
| MTCN (MFormer-HR) [33] | IN-21k+K400 +VGG-Sound | 70.7 | 62.1 | 49.6 |
| Omnivore (Swin-B) [22] | IN21k+IN-1k +K400+SUN | 69.5 | 61.7 | 49.9 |
| MeMViT [76] | K600 | 71.4 | 60.3 | 48.4 |
| LAVILA (TSF-L) | WIT+Ego4D | 72.0 | 62.9 | **51.0** |

Table 9. **The performance of action recognition on EK-100**. We report top-1 accuracy on verb, noun, and action. LAVILA outperforms all prior works in terms of action-level top-1 accuracy.

### D.4. Action Recognition on EGTEA

We fine-tune the pre-trained model on EGTEA for 100 epochs using SGD with a momentum of 0.9 and weight decay of 5e-4. We use cosine annealing with warmup, where the base learning rate starts from 1e-6, linearly increases to a peak of 3e-3 in the first epoch and then gradually decreases to 1e-5 following a half-wave cosine schedule. We drop the linear projection head and attach a 106-dim head for classification. For LAVILA, we train the classification head with $1\times$ base learning rate and the backbone with $0.1\times$. For visual-only video model pre-trained on Kinetics, we use $1\times$ base learning rate for both the classification head and the backbone. We use a per-gpu batch size of 16 over 8 GPUs for TimeSformer-B and a per-gpu batch size of 4 over 32 GPUs for TimeSformer-L. We use a stochastic depth ratio of 0.1 in the backbone and a dropout of 0.5 before the classification head. We also use a label smoothing of 0.1.

For input, we randomly select a 32-frame video clip at a temporal stride of 2 (namely $16\times 2$) from each video at training time. We scale the short side of the video to 256 pixels and then take a $224\times 224$ crop. For data augmentation, we use standard `RandomResizedCrop (0.5, 1.0)` and `RandomHorizontalFlip(0.5)`. At testing time, we evenly take ten 32-frame clips through the full video. We scale the short side to 224 pixels and take three spatial crops along the longer axis per clip. The final predictions are averaged over all these crops.

### D.5. Action Recognition on EK-100

We fine-tune the pre-trained model on EK100 with a same training schedule as in EGTEA. The only exception is that we apply three classification heads for verb, noun, and action separately because we empirically observe that it speeds up convergence and performs slightly better than using a single action-level classification head.

For the input, we represent each video clip with 16 sampled frames at both training and testing time. At testing time, we take three spatial crops along the longer axis per clip and average the final predictions.

### D.6. Action Recognition on CharadesEgo

Following EgoVLP [39], we convert the task of action classification to that of video-text retrieval as follows: for each trimmed video clip with textual annotations, we consider it to be a valid video-text pair for training. Since CharadesEgo is a multi-class dataset, which means each trimmed video can be annotated with different classes, we treat any trimmed video clip with $N$ actions as $N$ individual video-text pairs. We use the same InfoNCE [48] loss. We fine-tune the pre-trained model on CharadesEgo using AdamW with $(\beta_1, \beta_2) = (0.9, 0.999)$ and weight decay of 0.01. We use cosine annealing with warmup, where the peak learning rate is set to be 3e-5. For input, we randomly select a 32-frame video clip at a stride of 2 from the *trimmed* video at training time and evenly sample 16 frames from the *untrimmed* video at testing time to calculate the video-level mAP. We finetune the model for 10 epochs and report the best performance.

### D.7. LAVILA for Third-person Video Pre-training

The pre-training recipe mostly follows the one in Appendix D.1 except that when constructing a batch of samples, we sample one more hard negative clip from the same video for each selected clip following [25].

When doing linear-probing evaluation, we keep the video encoder frozen, extract video feature and train a linear SVM on top. For each video clip in either HMDB-51 or UCF-101, we evenly take four 32-frame clips through the entire video. We scale the short side to 224 pixels and take the center crop per clip and pass through the frozen video encoder to get the final visual embedding. For each testing video, we average the prediction score from different clips. We use scikit-learn's LinearSVC and report the highest top-1 accuracy after sweeping the regularization parameter $C \in \{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 0.1, 1, 10^2, 10^3, 10^4\}$.

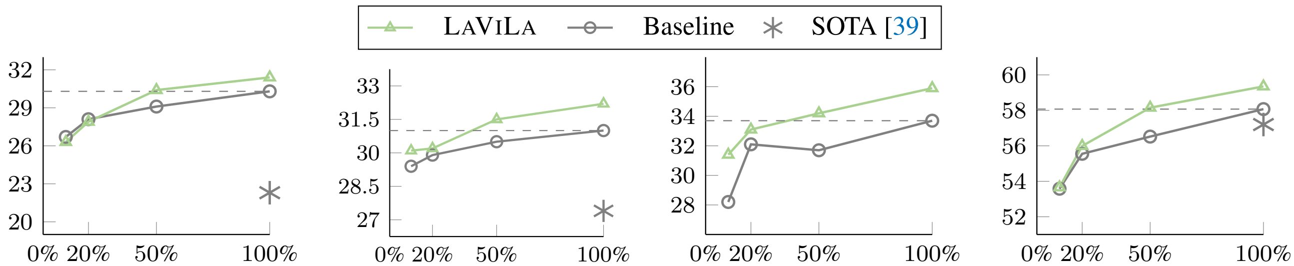
## E. Additional Results

**EK-100 CLS**. We compare LAVILA representation on EK-100 CLS in Table 9. We achieve state-of-the-art performance in terms of top-1 action accuracy. Note that the second best-performing Multiview Transformer [79] is pre-trained on WTS-60M which is not publicly available.

**More results on Semi-supervised Learning**. Following the setup in § 5.3, we provide more results in Figure 6 while replacing the backbone of LAVILA with TimeSformer-Large. We observe similar trends as § 5.3 where LAVILA outperforms the ground-truth-only baseline at all data points.

## F. Additional Ablations

**Improved Baseline on EK-100 MIR.** We present an improved baseline of video-language model pretrained on



(a) EK-100 MIR mAP. (b) EK-100 MIR nDCG. (c) EGTEA mean accuracy. (d) EgoMCQ Intra-video accuracy.

Figure 6. **More results of LaViLa in a semi-supervised setting where only a limited amount of narrations are given**. Both LaViLa and the baseline use a TimeSformer-Large as the visual encoder backbone. Comparing zero-shot performance of pre-training, LaViLa consistently outperforms the groundtruth-only baseline when 10, 20, 50,100% data is used.

| | EgoNCE | CLIP-init. | # frames | Avg. mAP | Avg. nDCG |
|---|---|---|---|---|---|
| Extracted RGB frames | | | | | |
| EgoVLP [39] | | | 4 | 15.5 | 22.1 |
| EgoVLP [39] | ✓ | | 4 | 16.6 | 23.1 |
| Videos (downsized to 480p) | | | | | |
| EgoVLP [39] | ✓ | | 4 | 22.3 | 27.4 |
| EgoVLP [39] | ✓ | | 16 | 23.6 | 27.9 |
| Our impl. | | | 4 | 24.1 | 28.0 |
| Our impl. | | ✓ | 4 | 24.7 | 28.4 |

Table 10. **Improved baseline** evaluted on EK-100 MIR. We observe that evaluting on videos directly improves the baseline noticeably. Using CLIP-pre-trained encoder weights introduces additional improvements. All gains shown in the paper are on top of this already strong baseline (last row).

Ego4D and evaluate it on EK-100 MIR in a zero-shot setting in Table 10. The initial baseline is video-language model with a TimeSformer-Base as visual encoder and a DistilBERT as textual encoder, proposed in EgoVLP [39]. First, we find that zero-shot evaluation on videos brings a noticeable improvement than on extracted RGB frames. Particularly, given the same EgoVLP+EgoNCE model, zero-shot retrieval can increase by 5.7% average mAP and 4.3% average nDCG repespectively. This is probably because frame extraction using ffmpeg's default parameter downgrades the image quality by a considerable amount. Second, under the same video-as-input evaluation protocol, our implementation with the same backbone (TimeSformer-Base + DistilBERT) using standard InfoNCE loss *without* EgoNCE, can achieve 24.1% and 28.0% average mAP and nDCG, better than the EgoVLP with EgoNCE. Third, if we pretrain the joint model using CLIP-pretrained models as the initial weights, the zero-shot retrieval result can be further boosted (+0.6% avg. mAP and +0.4% avg. nDCG), indicating that egocentric video representation can also benefit from large-scale image-text pre-training.

Starting from this improved baseline, we conduct more ablations on pretraining the video-langauge model in Table 11 as follows. We measure the performance by zero-shot average mAP and average nDCG on EK-100 MIR.

**Effect of weight initialization**. We study the effect of architectures and weight initialization in Table 11a. First, we observe that using the same architecture of TimeSformer-B, using CLIP-initialized weights pretrained on WebImageText (WIT) [49] works slightly better than using those supervised pretrained on ImageNet-21k [15, 61]. Second, if we replace the visual encoder with a ViT-Base model as in CLIP, the performance drops by 1.5% avg. mAP and 1.0% avg. nDCG, indicating the necessity of using spatial-temporal visual encoder for learning video-language tasks.

**Effect of batch size**. We study the effect of batch size of contrastive pre-training in Table 11b. The baseline method follows EgoVLP [39] and uses a total batch size of 512. We observe that the performance improves when increasing the batch size to 1,024. The improvment diminishes if we further increase the batch size to 2,048. Therefore, we use 1,024 as the default batch size to get our main results in § 5.1.

**Effect of projection dimension**. We compare different choices of the projection head's dimension in Table 11c. We can see that using 256 achieves the best performance compared to 128 or 512.

**Temperature in contrastive loss**. In Table 12, we study the effect of different temperatures in the contrastive loss (Eq 4). Note that we switch to a batch size of 1,024 based on the observation in Table 11b. We start with a learnable temperature of 0.07 following CLIP [49]. We can see that using a higher initial temperature $\tau_n$ for the pairs generated by Narrator achieves noticable gain over the one that uses the same initial temperature of 0.07 for both $\tau_r$ and $\tau_n$. We found that the within-batch accuracy during contrastive training for Narrator's pairs is significantly higher than the one for Rephraser's pairs. Our conjection is that the dual-encoders is more likely to overfit the Narrator's pairs. Therefore, we switch to a fixed temperature and find that using $\tau_r = \tau_n = 0.07$ works better than all other settings, such as learnable temperature.

| Vis. Enc. arch. | Vis. Enc. init. | Text Enc. | Text Enc. init. | avg mAP | avg. nDCG |
|---|---|---|---|---|---|
| TSF-B | IN-21K | DistilBERT | BC+Wiki | 24.1 | 28.0 |
| TSF-B | WIT | DistilBERT | BC+Wiki | 24.2 | **28.5** |
| ViT-B | WIT | CLIP-GPT | WIT | 23.2 | 27.4 |
| TSF-B | WIT | CLIP-GPT | WIT | **24.7** | 28.4 |

(a) **initialization**. IN-21K and WIT denote ImageNet-21k [15] and WebImageText [49]. BC+Wiki denotes BookCorpus+English Wikipedia on which BERT is pre-trained. Using CLIP-initialized weights works better than using those supervised pretrained on IK-21K.

| Batch size | Avg. mAP | Avg. nDCG |
|---|---|---|
| 512 | 24.7 | 28.4 |
| 1024 | **25.6** | **28.8** |
| 2048 | **25.6** | 28.5 |

(b) **Batch size**. Zero-shot performance improves when batch size increases from 512 to 1,024.

| Projection head | Avg. mAP | Avg. nDCG |
|---|---|---|
| Linear (128-d) | 24.1 | 27.8 |
| Linear (256-d) | **24.7** | **28.4** |
| Linear (512-d) | 24.5 | 28.1 |

(c) **Projection head**. Zero-shot performance is affected by the hidden dimension of the projection head. Empirically using 256 yields a best performance.

Table 11. **Ablations of dual-encoder**. We study how weight initialization (a), pre-training batch size (b), and project head dimension (c) affect the zero-shot performance of the dual-encoder on EK-100 MIR.

| $\tau_r$ | learn | $\tau_n$ | learn | Avg. mAP | Avg. nDCG |
|---|---|---|---|---|---|
| 0.07 | ✓ | n/a | n/a | 25.6 | 28.8 |
| 0.07 | ✓ | 0.07 | ✓ | 25.7 | 29.0 |
| 0.07 | ✓ | 0.10 | ✓ | 26.8 | 29.6 |
| 0.07 | ✓ | 0.10 | ✗ | 27.4 | 29.8 |
| 0.07 | ✗ | n/a | n/a | 26.0 | 29.0 |
| 0.07 | ✗ | 0.07 | ✗ | **29.5** | **31.1** |
| 0.07 | ✗ | 0.10 | ✗ | 27.4 | 29.8 |

Table 12. **Temperature in contrastive loss.** We observe that using a same fixed temperature for both NARRATOR's pairs and REPHRASER's pairs works better than all other settings.

## G. Qualitative Results

We provide more generated samples by our NARRATOR and REPHRASER in Figure 7. Note that our NARRATOR can generate reasonable captions from different views. For instance, Figure 7(d) illustrates that NARRATOR can describe the activities of both the camera wearer (starting with "C", which stands for "Camera wearer" in Ego4D) and the other person (starting with "O", which stands for "Observer" in Ego4D).

## H. Licenses

HMDB data is licensed under the CC BY 4.0 license and the data is available at `https://serre-lab.clps.brown.edu/resource/hmdb-a-large-human-motion-database/`.

The images in Figs. 2 to 4 and 7 are adapted from Ego4D videos. The video id (`$vid`) along with the start/end timestamp is provided below. The video can be viewed via the url `https://visualize.ego4d-data.org/$vid` (License is required for access).

- Figure 2: `1bfac46e-f957-4495-9583-dbd7fa683225`, `01:30:00-01:50:00`.
- Figure 3 (top): `06919917-76bc-4adc-b944-2a722f165513`, `00:00:08-00:00:10`.
- Figure 3 (bottom): `cf7c12db-1a9e-46d3-96d6-38174bbe373c`, `00:21:17-00:21:19`.
- Figure 4: `3c0dffd0-e38e-4643-bc48-d513943dc20b`, `00:00:12-00:00:14`.
- Figure 7 (a): `26054ab4-4967-47b5-9b6c-e8a62f9295e0`, `00:08:09-00:08:10`.
- Figure 7 (b): `3130e00e-873a-4afb-93a6-7b07f3cf6597`, `00:11:42-00:11:44`.
- Figure 7 (c): `def2e8dd-aaf7-467f-aa8f-46f654e6f4e0`, `00:09:08-00:09:09`.
- Figure 7 (d): `ab865129-78fa-47d4-8a50-ff8c5533246f`, `00:04:10-00:04:12`.
- Figure 7 (e): `58a01f3a-52ce-4024-ab3c-b179caf4dafd`, `00:28:43-00:28:45`.

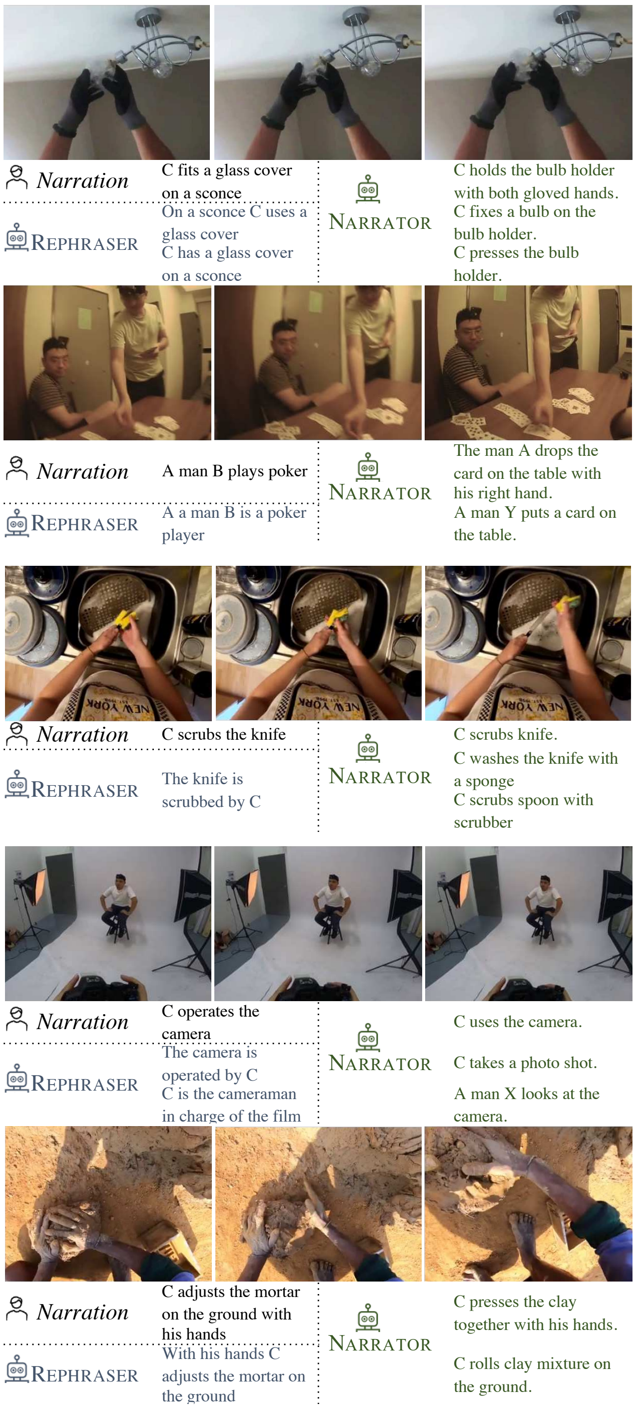



Figure 7. **More generated samples by our NARRATOR and REPHRASER on Ego4D**. NARRATOR generates new descriptions of the action taking place, potentially focusing on other objects or person being interacted with. REPHRASER not only changes the word order of the human narration but also diversifies it by using related verbs or nouns. Please refer to Appendix G for discussion.

## References


[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning. In *NeurIPS*, 2022.

[2] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *ICCV*, 2021.

[3] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[4] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *ICCV*, 2021.

[5] Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. A neural probabilistic language model. In *NIPS*, 2000.

[6] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, 2021.

[7] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *NeurIPS*, 2020.

[8] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015.

[9] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017.

[10] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *ECCV*, 2020.

[11] Jaemin Cho, Jie Lei, Hao Tan, and Mohit Bansal. Unifying vision-and-language tasks via text generation. In *ICML*, 2021.

[12] Jinwoo Choi, Gaurav Sharma, Manmohan Chandraker, and Jia-Bin Huang. Unsupervised and semi-supervised domain adaptation for action recognition from drones. In *WACV*, 2020.

[13] Ego4D Consortium. Egocentric live 4d perception (Ego4D) database: A large-scale first-person video database, supporting research in multi-modal machine perception for daily life activity. https://sites.google.com/view/ego4d/home. Accessed: 2022-11-22.

[14] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling egocentric vision: collection, pipeline and challenges for Epic-Kitchens-100. *IJCV*, 2022.

[15] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009.

[16] Karan Desai and Justin Johnson. Virtex: Learning visual representations from textual annotations. In *CVPR*, 2021.

[17] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, 2015.

[18] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020.

[19] Marzieh Fadaee, Arianna Bisazza, and Christof Monz. Data augmentation for low-resource neural machine translation. In *ACL*, 2017.

[20] Steven Y Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. A survey of data augmentation approaches for nlp. In *ACL Findings*, 2021.

[21] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. *NeurIPS*, 2013.

[22] 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.

[23] Ramsri Goutham Golla. High-quality sentence paraphraser using transformers in nlp. https://huggingface.co/ramsrigouthamg/t5-large-paraphraser-diverse-high-quality. Accessed: 2022-06-01.

[24] 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, 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, Ziwei Zhao, Yunyi Zhu, Pablo Arbelaez, David Crandall, Dima Damen, Giovanni Maria Farinella, Christian Fuegen, 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 *CVPR*, 2022.

[25] Tengda Han, Weidi Xie, and Andrew Zisserman. Temporal alignment networks for long-term video. In *CVPR*, 2022.

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[27] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 1997.

[28] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

[29] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *ICLR*, 2020.

[30] Xiaowei Hu, Zhe Gan, Jianfeng Wang, Zhengyuan Yang, Zicheng Liu, Yumao Lu, and Lijuan Wang. Scaling up vision-language pre-training for image captioning. In *CVPR*, 2022.

[31] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, 2021.

[32] Chen Ju, Tengda Han, Kunhao Zheng, Ya Zhang, and Weidi Xie. Prompting visual-language models for efficient video understanding. In *ECCV*, 2022.

[33] Evangelos Kazakos, Jaesung Huh, Arsha Nagrani, Andrew Zisserman, and Dima Damen. With a little help from my temporal context: Multimodal egocentric action recognition. In *BMVC*, 2021.

[34] Dan Kondratyuk, Liangzhe Yuan, Yandong Li, Li Zhang, Mingxing Tan, Matthew Brown, and Boqing Gong. Movinets: Mobile video networks for efficient video recognition. In *CVPR*, 2021.

[35] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *ICCV*, 2011.

[36] Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L Berg, Mohit Bansal, and Jingjing Liu. Less is more: Clipbert for video-and-language learning via sparse sampling. In *CVPR*, 2021.

[37] Yin Li, Miao Liu, and James M Rehg. In the eye of beholder: Joint learning of gaze and actions in first person video. In *ECCV*, 2018.

[38] Yanghao Li, Tushar Nagarajan, Bo Xiong, and Kristen Grauman. Ego-exo: Transferring visual representations from third-person to first-person videos. In *CVPR*, 2021.

[39] Kevin Qinghong Lin, Alex Jinpeng Wang, Mattia Soldan, Michael Wray, Rui Yan, Eric Zhongcong Xu, Difei Gao, Rongcheng Tu, Wenzhe Zhao, Weijie Kong, Hongfa Cai Chengfei, Wang, Dima Damen, Bernard Ghanem, Wei Liu, and Mike Zheng Shou. Egocentric video-language pre-training. In *NeurIPS*, 2022.

[40] Ziyi Lin, Shijie Geng, Renrui Zhang, Peng Gao, Gerard de Melo, Xiaogang Wang, Jifeng Dai, Yu Qiao, and Hongsheng Li. Frozen clip models are efficient video learners. In *ECCV*, 2022.

[41] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *NeurIPS*, 2019.

[42] Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, Jason Li, Taroon Bharti, and Ming Zhou. Univl: A unified video and language pre-training model for multimodal understanding and generation. *arXiv preprint arXiv:2002.06353*, 2020.

[43] Maluuba. nlg-eval. https://github.com/Maluuba/nlg-eval. Accessed: 2022-06-01.

[44] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *CVPR*, 2020.

[45] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *ICCV*, 2019.

[46] Arsha Nagrani, Paul Hongsuck Seo, Bryan Seybold, Anja Hauth, Santiago Manen, Chen Sun, and Cordelia Schmid. Learning audio-video modalities from image captions. In *ECCV*, 2022.

[47] Bolin Ni, Houwen Peng, Minghao Chen, Songyang Zhang, Gaofeng Meng, Jianlong Fu, Shiming Xiang, and Haibin Ling. Expanding language-image pretrained models for general video recognition. In *ECCV*, 2022.

[48] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[49] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[50] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 2019.

[51] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*, 2020.

[52] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

[53] Anna Rohrbach, Atousa Torabi, Marcus Rohrbach, Niket Tandon, Christopher Pal, Hugo Larochelle, Aaron Courville, and Bernt Schiele. Movie description. *IJCV*, 2017.

[54] Christoph Schuhmann, Romain Beaumont, Cade W Gordon, Ross Wightman, Theo Coombes, Aarush Katta, Clayton Mullis, Patrick Schramowski, Srivatsa R Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. Laion-5b: An open large-scale dataset for training next generation image-text models. In *NeurIPS D&B*,

2022.
[55] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *ACL*, 2016.
[56] Paul Hongsuck Seo, Arsha Nagrani, Anurag Arnab, and Cordelia Schmid. End-to-end generative pretraining for multimodal video captioning. In *CVPR*, 2022.
[57] Gunnar A Sigurdsson, Abhinav Gupta, Cordelia Schmid, Ali Farhadi, and Karteek Alahari. Actor and observer: Joint modeling of first and third-person videos. In *CVPR*, 2018.
[58] 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.
[59] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. *NeurIPS*, 2016.
[60] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.
[61] Andreas Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your vit? data, augmentation, and regularization in vision transformers. *TMLR*, 2022.
[62] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. In *ICLR*, 2020.
[63] Swathikiran Sudhakaran, Sergio Escalera, and Oswald Lanz. Lsta: Long short-term attention for egocentric action recognition. In *CVPR*, 2019.
[64] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *ICCV*, 2019.
[65] Maria Tsimpoukelli, Jacob L Menick, Serkan Cabi, SM Eslami, Oriol Vinyals, and Felix Hill. Multimodal few-shot learning with frozen language models. In *NeurIPS*, 2021.
[66] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.
[67] Ashwin K Vijayakumar, Michael Cogswell, Ramprasath R Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. Diverse beam search: Decoding diverse solutions from neural sequence models. *arXiv preprint arXiv:1610.02424*, 2016.
[68] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *CVPR*, 2015.
[69] Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, et al. Image as a foreign language: Beit pretraining for all vision and vision-language tasks. *arXiv preprint arXiv:2208.10442*, 2022.
[70] William Yang Wang and Diyi Yang. That's so annoying!!!: A lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using# petpeeve tweets. In *EMNLP*, 2015.
[71] Xiaohan Wang, Linchao Zhu, Heng Wang, and Yi Yang. Interactive prototype learning for egocentric action recognition. In *ICCV*, 2021.
[72] Jason Wei and Kai Zou. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. In *EMNLP*, 2019.
[73] Jason Weston, Samy Bengio, and Nicolas Usunier. Large scale image annotation: learning to rank with joint word-image embeddings. *Machine learning*, 2010.
[74] John Wieting and Kevin Gimpel. Paranmt-50m: Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. In *ACL*, 2018.
[75] Michael Wray, Diane Larlus, Gabriela Csurka, and Dima Damen. Fine-grained action retrieval through multiple parts-of-speech embeddings. In *ICCV*, 2019.
[76] Chao-Yuan Wu, Yanghao Li, Karttikeya Mangalam, Haoqi Fan, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Memvit: Memory-augmented multiscale vision transformer for efficient long-term video recognition. In *CVPR*, 2022.
[77] Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer. Videoclip: Contrastive pre-training for zero-shot video-text understanding. In *EMNLP*, 2021.
[78] Hongwei Xue, Yuchong Sun, Bei Liu, Jianlong Fu, Ruihua Song, Houqiang Li, and Jiebo Luo. Clip-vip: Adapting pre-trained image-text model to video-language representation alignment. In *ICLR*, 2023.
[79] Shen Yan, Xuehan Xiong, Anurag Arnab, Zhichao Lu, Mi Zhang, Chen Sun, and Cordelia Schmid. Multiview transformers for video recognition. In *CVPR*, 2022.
[80] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. CoCa: Contrastive captioners are image-text foundation models. *TMLR*, 2022.
[81] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, Ce Liu, Mengchen Liu, Zicheng Liu, Yumao Lu, Yu Shi, Lijuan Wang, Jianfeng Wang, Bin Xiao, Zhen Xiao, Jianwei Yang, Michael Zeng, Luowei Zhou, and Pengchuan Zhang. Florence: A new foundation model for computer vision. *arXiv preprint arXiv:2111.11432*, 2021.
[82] Rowan Zellers, Ximing Lu, Jack Hessel, Youngjae Yu, Jae Sung Park, Jize Cao, Ali Farhadi, and Yejin Choi. Merlot: Multimodal neural script knowledge models. *NeurIPS*, 2021.
[83] Andy Zeng, Adrian Wong, Stefan Welker, Krzysztof Choromanski, Federico Tombari, Aveek Purohit, Michael Ryoo, Vikas Sindhwani, Johnny Lee, Vincent Vanhoucke, and Pete Florence. Socratic models: Composing zero-shot multimodal reasoning with language. In *ICLR*, 2023.
[84] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *CVPR*, 2022.
[85] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *NeurIPS*, 2015.
[86] Luowei Zhou, Chenliang Xu, and Jason J Corso. Towards automatic learning of procedures from web instructional videos. In *AAAI*, 2018.
[87] Xingyi Zhou, Rohit Girdhar, Armand Joulin, Philipp Krähenbühl, and Ishan Misra. Detecting twenty-thousand classes using image-level supervision. In *ECCV*, 2022.
[88] Linchao Zhu and Yi Yang. Actbert: Learning global-local video-text representations. In *CVPR*, 2020.
[89] Xizhou Zhu, Jinguo Zhu, Hao Li, Xiaoshi Wu, Hongsheng Li, Xiaohua Wang, and Jifeng Dai. Uni-perceiver: Pre-

training unified architecture for generic perception for zero-shot and few-shot tasks. In *CVPR*, 2022.