Meta-Personalizing Vision-Language Models to Find Named Instances in Video

Chun-Hsiao Yeh[1,3*] Bryan Russell[3] Josef Sivic[2,3] Fabian Caba Heilbron[3†] Simon Jenni[3†]
[1]University of California, Berkeley [2]CIIRC CTU [3]Adobe Research
daniel_yeh@berkeley.edu {brussell,inr03127,caba,jenni}@adobe.com

Abstract

Large-scale vision-language models (VLM) have shown impressive results for language-guided search applications. While these models allow category-level queries, they currently struggle with personalized searches for moments in a video where a specific object instance such as "My dog Biscuit" appears. We present the following three contributions to address this problem. First, we describe a method to meta-personalize *a pre-trained VLM,* i.e.*, learning how to learn to personalize a VLM at test time to search in video. Our method extends the VLM's token vocabulary by learning novel word embeddings specific to each instance. To capture only instance-specific features, we represent each instance embedding as a combination of shared and learned global category features. Second, we propose to learn such personalization without explicit human supervision. Our approach automatically identifies moments of named visual instances in video using transcripts and vision-language similarity in the VLM's embedding space. Finally, we introduce This-Is-My, a personal video instance retrieval benchmark. We evaluate our approach on This-Is-My and DeepFashion2 and show that we obtain a 15% relative improvement over the state of the art on the latter dataset.*

1. Introduction

The recent introduction of large-scale pre-trained vision-language models (VLMs) has enabled many new vision tasks, including zero-shot classification and retrieval [14, 18, 22], image/video generation [12, 24, 25, 27, 28, 32], or language-guided question answering [1, 19, 42]. It is now possible to search not only for specific object categories (*e.g.*, dogs) but also for more specific descriptions of both the object and scene attributes (*e.g.*, "A small white dog playing at the dog park"). However, we often do not want to search for just any example of a generic category but instead to find a specific instance. For example, a user might want to search their personal video library for all the scenes that show their dog "Biscuit grabbing a pink frisbee", as illustrated in Figure 1. Since VLMs do not have a representation of "Biscuit," such queries are beyond the capabilities

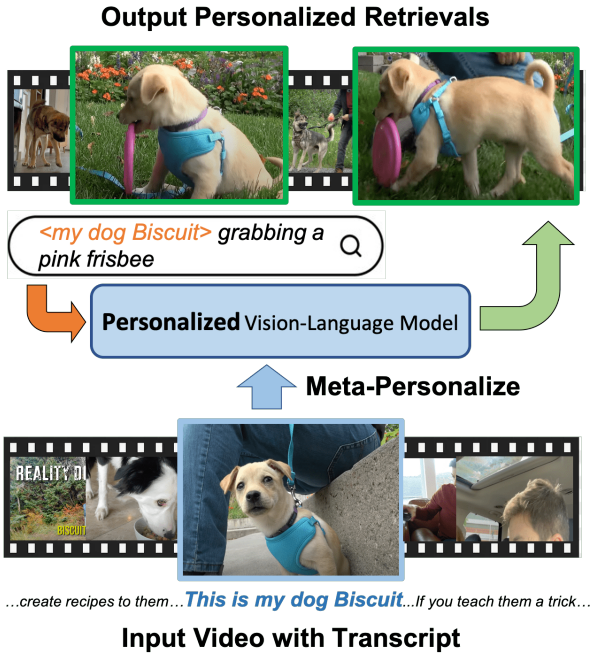

Figure 1. **Meta-Personalized Vision-Language Model (VLM) to Retrieve Named Instances in Video.** Given a video where a user-specific instance, *e.g.*, "My dog Biscuit" is mentioned, our method automatically learns a representation for the user-specific instance in the VLM's text input space. The personalized VLM can then be used to retrieve the learned instance in other contexts through natural language queries, *e.g.*, `<my dog Biscuit> grabbing a pink frisbee`. This result is enabled by meta-personalizing the VLM on a large-scale dataset of narrated videos by pre-learning shared global category tokens (in this example for the category of 'dogs'), which are then easily personalized to user-specific instances from only a few user-given training examples.

[†]Equal advising.
[*]Work done during CHY's summer internship at Adobe Research.
[2]Czech Institute of Informatics, Robotics and Cybernetics at the Czech Technical University in Prague.

of off-the-shelf VLMs.

Recent work [5] proposed a method to extend the language encoder's vocabulary with a newly learned token that represents a specific personal instance to address this issue. While this approach enables language-guided search for personal instances by placing the learned tokens in the query prompt, their solution assumes a collection of manually annotated images showing the individual instance in various contexts for successful token learning. For this approach to work in practice, a user must manually annotate all their important personal instances in various contexts, such that the instance representation does not capture nuisance features, *e.g.*, the background. We thus identify two key challenges: 1) collecting personal instance examples without explicit human labeling and 2) learning a generalizable object-centric representation of personal instances from very few examples.

The contributions of this work are three-fold. As our first contribution, we propose a method to automatically identify important personal instances in videos for personalizing a vision-language model without explicit human annotations. Indeed, people often record and refer to personal items or relationships in videos found online. Our approach is thus to identify mentions of personal instances in a video automatically and leverage these moments to build a set of personal instances for training. To this end, we extract the transcripts of videos using speech-to-text models and find candidate moments by looking for occurrences of "this is my *" or similar possessive adjective patterns. The symbol * in this example could represent a single word or sequence of words describing the instance (*e.g.*, *= "dog Biscuit"). We then use vision-language similarity to filter non-visual examples and to find additional occurrences in the video for training For example, we found more than six thousand named instances in 50K videos randomly sampled from the Merlot Reserve dataset [44]. We call the resulting collection of named instances in videos the *This-Is-My* dataset.

As our second contribution, we propose a novel model and training procedure to learn text tokens representing the named instances in video from possibly very few and noisy training examples. Our method represents each instance with learned tokens and models each token as a linear combination of a set of pre-learned category-specific features shared across different instances. This set of shared category-specific features (similar to object attributes) improves the generalization of our method by preventing the instance representations from capturing nuisance features (*e.g.*, the scene background). Furthermore, we show how to pre-train and adapt the shared category features using a large set of automatically collected *This-Is-My* examples, further improving our model's few-shot personalization performance at test-time. We call this pre-training of shared category features *meta-personalization*. In contrast to prior work [5], our method does not require training additional neural network models and requires only the optimization of a contrastive learning objective.

As our final contribution, we demonstrate and evaluate our model on an existing fashion item retrieval benchmark, DeepFashion2, and our new challenging *This-Is-My* video instance retrieval dataset[4] depicting specific object instances across different videos and contexts. Our experiments demonstrate that our method outperforms several baselines and prior approaches on these challenging language-guided instance retrieval tasks.

2. Related Work

Vision-Language Models for Video Retrieval. Vision-language foundational models [14, 18, 22] have been successful for zero-shot and other diverse video tasks, such as video question answering [19, 42], language-video grounding [13, 37, 41], and text-to-video retrieval [11, 16, 17, 20, 21, 39, 40]. These models have a powerful representation that transfers well to the video domain to achieve competitive performance on video-language tasks. Our approach builds on these powerful representations to retrieve specific named instances in video.

Personalized Concept Learning. Adapting a model to learn a user-specific representation has been a significant topic in machine learning research, including recommendation systems [2, 3] and federated learning [15]. Relevant to us are recent approaches for adapting vision-language models to object instances. PALAVRA [5] proposes a learning scheme that appends a learnable token for a new personalized concept to the token embedding of the input text prompt. This learned representation helps to preserve the personalized concept. DualPrompt [36] introduces a framework that learns a small set of prompts to emphasize more specific concepts without forgetting the learned concepts in the pre-trained foundational model. There have also been works that have extended this personalized concept learning to image generation [4, 8, 26]. While these approaches adapt a vision-language model to personal instances, they perform the adaptation independently for each instance and do not "meta-train" for the personalization task for improved few-shot learning as we do in this work.

Fine-tuning and Test-time Adaptation. Fine-tuning is a common strategy to adapt a pretrained model for a specific downstream task by transferring the source model to a target domain. Recent works on vision-language model tuning include CLIP-Adapter [9] that proposes to conduct fine-tuning with an extra bottleneck layer while freezing the pre-trained CLIP model. WiSE-FT [38] resolves the distribution shift caused by fine-tuning and ensembles the weights of the original and fine-tuned model to increase ro-

[4] Available at https://danielchyeh.github.io/metaper/

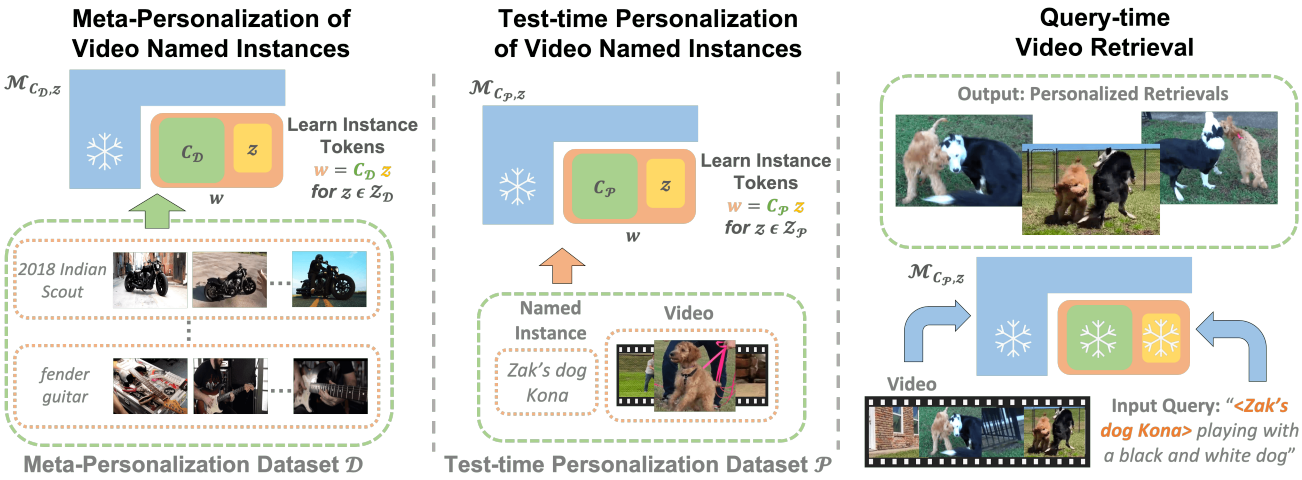

Figure 2. **Overview of our Personalized Vision-Language Model.** Our model augments a frozen VLM (blue) with novel personal instance tokens $w = Cz$ (orange) that are a combination of global category features C (green) with instance-specific weights $z \in \mathcal{Z}$ (yellow). Our approach for personalized instance retrieval has three stages. First, we pre-learn global category features $C_\mathcal{D}$ on a large set of automatically mined named personal instances in videos. We call this process Meta-Personalization (left). In the second step (middle), we adapt the meta-personalized category features $C_\mathcal{D}$ at test-time and learn novel instance weights $z \in \mathcal{Z}_\mathcal{P}$ to represent a user's personal instances via $w = C_\mathcal{P} z$. Finally (right), we leverage the (frozen) personalized instance tokens w in natural language queries at query time.

bustness. Prior test-time adaption works [31, 34, 35] fine-tune on target data without information from the source data. Our approach leverages test-time adaptation for updating our meta-personalized model to user-specific data.

Meta-Learning. We draw inspiration from meta-learning ("learning to learn") [6, 7, 43], which enables models to quickly adapt to new tasks by learning on a diverse set of tasks. Hence, given only a handful of novel training examples, the model can be adapted to novel tasks. Our approach borrows the idea of meta-learning to meta-personalize a model by learning global category features from a large video database. We then adapt the global features during test-time training from only few examples of user-specific instances to enable query-time retrieval.

3. (Meta-)Personalization of Named Instances

Our goal is to learn representations of personal items in video that enable retrieval through natural language queries. To achieve this goal, we have to address the key challenge of adapting a model with one or few examples of a named instance. To address this challenge, we propose a *meta-personalization* approach that learns to personalize given a large corpus of named instances mined from videos with transcriptions. We illustrate our approach in Figure 2.

In the first step (Figure 2 (left)), we mine automatically a collection $\mathcal{D}$ of named instances from videos with transcriptions for meta-personalization. We use this collection to train a proposed model $\mathcal{M}_{C,z}$ that includes global category features C and instance-specific parameters z. The global category features C are lightweight and shared across all instances. Given a natural language query u and video v, the model returns a score $\mathcal{M}_{C,z}(u, v)$. During meta-personalization, given a training loss $\mathcal{L}$, we jointly optimize the loss over the global category features C and instance parameters $\mathcal{Z}$ for each named instance in the collection $\mathcal{D}$,

$$(C_\mathcal{D}, \mathcal{Z}_\mathcal{D}) \in \arg\min_{(C,\mathcal{Z})} \sum_{z \in \mathcal{Z}} \mathcal{L}(C, z). \quad (1)$$

Note that here the instance-specific parameters $\mathcal{Z}_\mathcal{D}$ learnt via (1) are discarded while the global category features $C_\mathcal{D}$ are kept as the meta-personalized part of the model. The global category features $C_\mathcal{D}$ capture information shared across instances relevant to the personalization task.

In the second step (Figure 2 (middle)), we are given a set of named video instances $\mathcal{P}$ (*e.g.*, automatically mined from someone's personal video library) and wish to perform test-time personalization of the model to this person's instances. Here each instance is represented by only one or few examples. In this step, we optimize the training loss $\mathcal{L}$ over the global category features C and the set of instance parameters $\mathcal{Z}$ for all instances in the personal set $\mathcal{P}$ starting from the pre-trained global category features $C_\mathcal{D}$ and random $\mathcal{Z}$. We obtain the personalized model parameters $(C_\mathcal{P}, \mathcal{Z}_\mathcal{P})$ as

$$(C_\mathcal{P}, \mathcal{Z}_\mathcal{P}) \in \arg\min_{(C,\mathcal{Z})} \sum_{z \in \mathcal{Z}} \mathcal{L}(C, z). \quad (2)$$

We now keep both $C_\mathcal{P}$ and $\mathcal{Z}_\mathcal{P}$.

In the final step (Figure 2 (right)), we perform retrieval over a potentially large dataset using the test-time personalized model $\mathcal{M}_{C_\mathcal{P},z}$ (where $z \in \mathcal{Z}_\mathcal{P}$). We next describe

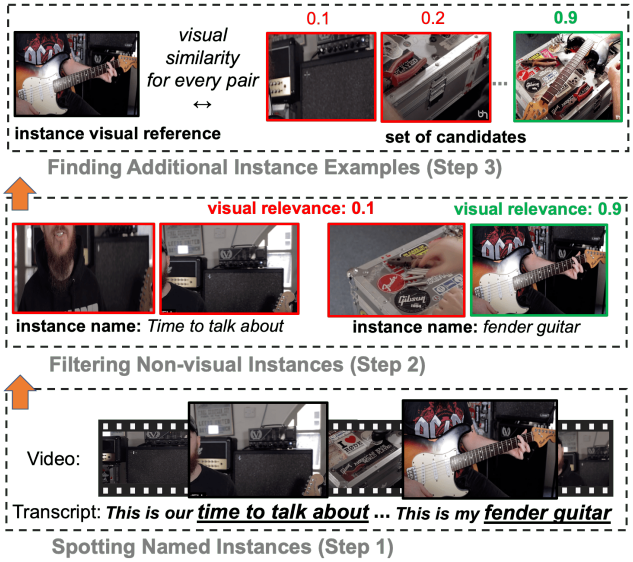

Figure 3. **Automatic mining of named instances in video for meta-personalization.** Our automatic mining pipeline includes three steps (from bottom to top). **Step 1** finds named instances via string-matching of possessive patterns in video transcripts. **Step 2** filters non-visual instances using text-to-visual relevance between the instance name and the video shots neighboring the named instance. Finally, **Step 3** retrieves additional shots with high visual similarity to the instance reference shot.

how we automatically mine the named instances in video $\mathcal{D}$ for meta-personalization (Sec. 3.1), and our full model $\mathcal{M}$ for query-time retrieval and loss $\mathcal{L}$ for meta-personalization and test-time personalization (Sec. 3.2).

3.1. Automatic Mining of Named Instances in Video

To identify personal instances without explicit supervision, we leverage a collection of videos from the web along with their corresponding time-aligned transcripts. These transcripts can be automatically generated through speech-to-text models [23, 33]. We now describe a procedure to mine these data for a set of moments (*i.e.*, a collection of video shots) depicting a referred personal instance in the transcript without the need for manual annotation. We will use these moments for training a meta-personalization model (Sec. 3.2).

Spotting Named Instances. Here our goal is to find moments where candidate personal instances are mentioned in videos. We do so by searching for possessive text patterns[5] such as `"This is my *"` in a corpus of time-aligned video transcripts. This string-matching process outputs a list of candidate instance names $*$ associated with video timestamps t^*. In practice, we keep up to four words after a possessive text pattern is matched based on text-visual similarity (see supp. for details). That way, we can retrieve simple named instances such as `This is my dog` ($*$ =`dog`) but also complex ones like `This is my favorite CHANEL classic handbag` ($*$ =`CHANEL classic handbag`). Note also that a single video might include multiple string matches; for instance, the example video illustrated in Figure 3 (Step 1) includes two matches: `"This is our time to talk about"` at time `1:30` and `"This is my fender guitar"` at time `3:25`.

Filtering Non-visual Instances. The previous spotting step only searches for potential instances using the transcript, yielding many string matches that are *non-visual*, *i.e.*, the strings do not describe the visible content in the video. Here we aim to filter out these non-visual instances. We do so by computing the text-to-visual relevance between the instance name (*e.g.*, `fender guitar`) and the neighboring shots around the time when the instance is mentioned. We add neighboring shots to cover cases where the named instances are shown just before or after they are mentioned. Concretely, given a sequence of m video shots $S = [s_1, \dots, s_m]$ automatically extracted with [30], we find the shot s_{t^*} that overlaps with t^* (the time when the instance was mentioned). Next, we form a set of candidate visual references $S_{t^*} = [s_{t^*-1}, s_{t^*}, s_{t^*+1}]$ comprising a window of shots that are previous and subsequent to s_{t^*}. We then compute text-to-visual relevance scores using CLIP [22] encoders. This encoding process yields L_2-normalized embeddings $f_l(*)$ for the named instance and $f_v(s_i)$ for each shot $s_i \in S_{t^*}$. We compute $f_v(s_i)$ by averaging the visual embeddings of all frames in the corresponding shot. Finally, we compute the cosine similarity between every $(f_l(*), f_v(s_i))$ pair and retain a visual reference shot s^* if the highest cosine similarity is greater than 0.3. This filtering step outputs a cleaned set of named instances with a corresponding visual reference. Figure 3 (Step 2) illustrates how we prune out non-visual matches such as `time to talk about`. In contrast, visual instances such as `fender guitar` are kept and matched with a visual reference.

Finding Additional Instance Shots. Since frames from a single video shot provide only limited variability in the instance appearance and could thus limit the learning, we aim to recover other shots from the video where that instance appears. We leverage CLIP's visual encoder to compute the visual similarity between the instance's reference shot s^* and every shot $s_i \in S$. We extract an embedding $f_v(s^*)$ for the reference shot s^* and an embedding $f_v(s_i)$ for each shot s_i. Similar to the non-visual filtering step, we average the CLIP embeddings of each frame belonging to a shot. Then, we compute the cosine similarity between the embeddings for the reference shot and every candidate shot in the video. We keep the shots whose cosine similarity with the refer-

[5] **List of possessive text patterns:** `<this is my>`, `<this is our>`, `<this is his>`, `<this is her>`, `<this is their>`, `<these are my>`, `<these are our>`, `<these are his>`, `<these are her>`, `<these are their>`

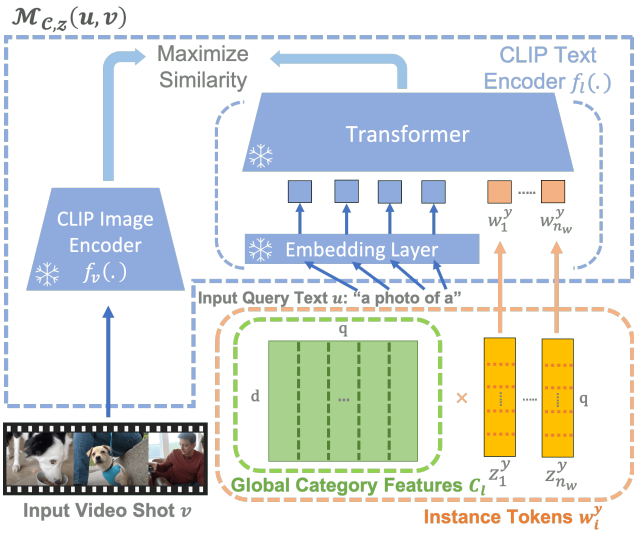

Figure 4. **Model Overview.** Our model $\mathcal{M}_{C,z}$ extends CLIP's language input vocabulary with n_w novel instance-specific tokens $w_i^y = C_l z_i^y$, which we model as a linear combination of meta-personalized category features C_l with weights z_i^y. Note that the vision and language encoders are frozen during this process.

ence is greater than 0.9. Figure 3 (Step 3) illustrates the output of this final step in the automatic mining pipeline. Our mining algorithm allow us to retrieve additional shots for the instance `fender guitar`. Now the instance examples not only include a clean close-up of the guitar, but also shots where the guitar is being played or held by its owner.

3.2. Learning Personal Instance Representations

The result of the mining procedure described above is a dataset consisting of a set of video shots $\mathcal{D} = \{s_1, \ldots, s_n\}$ and corresponding instance IDs $Y = \{y_1, \ldots, y_n\}$, where $y_i = y_j$ if s_i and s_j are video shots that are assumed to contain the same instance. We now describe how we use these data to learn representations of the collected instances.

Our approach is to leverage a large-scale pre-trained vision-language model (CLIP [22]) and augment its language encoder with a set of novel personal instance tokens. Let $f_v(s)$ be the output of the visual encoder for shot s (computed as the average over the frame embeddings) and let $f_l(u)$ be the output of the language encoder for a natural language input $u = [v_1, \ldots, v_m]$ of m token embeddings, where $v_i \in \mathbb{R}^d$ denote learned token embeddings of the pre-training vocabulary (positional embeddings are included but omitted from the notation). We propose to extend this vocabulary with novel personal instance tokens. Concretely, our approach introduces a set $w^y = \{w_i^y\}_{i=1}^{n_w}$ of n_w new tokens that represent a personal instance $y \in Y$.

To learn these tokens and to perform personalized retrieval at test time, we construct natural language personalized queries as

$$\hat{u}^p = [p_1, \ldots, p_{k-1}, w_1^y, \ldots, w_{n_w}^y, p_{k+1}, \ldots, p_m], \quad (3)$$

where p_i are token embeddings of a chosen prompt $p = [p_1, \ldots, p_m]$. During training, the prompt p corresponds to a random template of the form `[An image of *]`, `[* can be seen in this photo]`, `[There is * in this image]`, etc., and k denotes the position of the * placeholder for the instance tokens.

Instances as Combinations of Category Features. Since learning personal instance tokens from possibly very few examples runs the danger of overfitting (*e.g.*, to nuisance features in the background), we propose to parameterize them as

$$w_i^y = C_l z_i^y \in \mathbb{R}^{d \times 1}, \quad (4)$$

where $z_i^y \in \mathbb{R}^{q \times 1}$ is a vector of learnable weights specific to each instance and $C_l \in \mathbb{R}^{d \times q}$ is a matrix of learnable global category features, which are shared for all instances belonging to the same object category $l \in \mathcal{Y}$ (*e.g.*, $\mathcal{Y} = \{\text{car}, \text{person}, \text{dog}, \ldots\}$) and constitute the set $C_\mathcal{D} = \{C_l\}_{l \in \mathcal{Y}}$. We illustrate the model in Figure 4.

We can think of the columns of C_l as representing a set of shared category features and the final instance token w_i^y as a linear combination of these features with weights z_i^y. Our aim is that only category-specific features are captured during training and irrelevant features (*e.g.*, about the background) are discarded. Intuitively, the columns of C_l could correspond to attributes of an object category, *e.g.*, if we were to learn "car" features, these could capture their color, brand, type, or age, to name a few. To identify which category matrix C_l to use for an instance y, we rely on 0-shot classification using vision-language similarity between instance shots and a generic prompt for the category $l \in \mathcal{Y}$.

Contrastive Personal Token Learning. We propose a contrastive learning objective to learn the personal instance tokens w^y using a set of video shots containing the instance y. To this end, let $\psi_i := f_v(s_i)$ be an encoding of a video shot s_i, *i.e.*, the average frame encoding of all frames in the shot, and let $\phi_i := f_l(\hat{u}_i^p)$ denote the language encoding of a corresponding personalized query. We learn the novel tokens w^y and shared category features C_l by optimizing two contrastive objectives: a language-language contrastive objective $\mathcal{L}_l$ and a vision-language contrastive objective $\mathcal{L}_{vl}$. The language-language objective is given by

$$\mathcal{L}_l = \sum_{i \in \mathcal{B}} \sum_{j \neq i \in \mathcal{B}} -\mathbb{1}\{y_i = y_j\} \log \left(\frac{d(\phi_i, \phi_j)}{\sum_{k \neq i \in \mathcal{B}} d(\phi_i, \phi_k)} \right), \quad (5)$$

where $d(a, b) := \exp\left(\frac{1}{\lambda} \frac{a^\intercal b}{\|a\|_2 \|b\|_2}\right)$ measures the similarity between the feature vectors a and b, $\lambda = 0.1$ is a temperature parameter, and $\mathcal{B}$ is a randomly sampled training mini-batch. The vision-language objective is similarly defined as

Figure 5. **Examples from *This-Is-My* { Meta-Personalization $\mathcal{D}$ (top) vs Test-time personalization $\mathcal{P}$ (bottom-left) vs Query-time $\mathcal{Q}$ (bottom-right)} datasets.** In the Query-time dataset (bottom-right), we design a challenging video instance retrieval task. For example, in (a) the named instance (*i.e.*, Alex's piano) is in the background and is barely visible and in (b) the background scenes in the query-time dataset (bottom-right) are completely different from the test-time personalization dataset (bottom-left) depicting the same named instance.

$$\mathcal{L}_{vl} = \sum_{i,j \in \mathcal{B}} -\mathbb{1}\{y_i = y_j\} \log \left(\frac{d\left(\phi_i, \psi_j\right)}{\sum_{k \in \mathcal{N}} d\left(\phi_i, \psi_k\right)} \right), \quad (6)$$

with the set of negative examples $\mathcal{N}$ comprising both other examples in the batch $\mathcal{B}$ and non-instance shots from the videos containing the named instances, *i.e.*, shots that have low vision-language similarity. The loss is low when the encodings for video shots and personalized queries with the same instance ID are more similar than to other queries. Including non-instance segments as negatives can help discard non-instance features such as scene background.

To further constrain the learning toward category-specific attributes, we include a loss that maximizes the similarity between a personal instance query and a generic category query. Concretely, let c_l be a category query embedding for category l (*e.g.*, "An image of a [car]") to which instance y belongs. We then include the following category-anchoring loss

$$\mathcal{L}_c = -\sum_{i \in \mathcal{B}} \frac{{c_l}^\intercal \phi_i}{\|c_l\|_2 \|\phi_i\|_2}. \quad (7)$$

To summarize, our training loss $\mathcal{L}$ (see Equations 1 and 2) for meta- and test-time personalization is given by

$$\mathcal{L} = \mathcal{L}_l + \mathcal{L}_{vl} + \lambda_c \mathcal{L}_c, \quad (8)$$

where $\lambda_c = 0.5$, controls the amount of category anchoring.

4. *This-Is-My* Dataset

Our *This-Is-My* dataset comprises three subsets for meta-personalization, test-time personalization, and querying. We describe each subset next.

Meta-Personalization Dataset $\mathcal{D}$. We gather this subset with the automatic mining algorithm introduced in Section 3.1. We leverage the Merlot Reserve dataset [44], which contains more than 20 million videos from YouTube and their corresponding time-aligned transcripts. In practice, we start from a subset of $50K$ randomly sampled Merlot Reserve videos. We spot 6058 named instances using various text possessive templates. Our visual filtering step removes 52% of instances, generating a total of 2908 named instances with a visual reference. Finally, we mine additional samples for each instance, yielding a total of 49256 instance samples. This subset includes a wide variety of visual concepts, ranging from common objects such as bikes to rare concepts such as toaster. While we design our mining pipeline to minimize noise, it is limited by CLIP's capability to distinguish between similar object instances. For instance, while we find several samples of the `fender guitar` instance, it includes other shots that do not correspond to the aforementioned guitar (Figure 5 (top)). Nevertheless, we will show empirically that this subset is still useful for meta-personalization purposes (Section 5).

Test-time Personalization Dataset $\mathcal{P}$. Our goal is to create a test dataset that recreates the scenario where a person wants to find when their personal instances appear in their video collection. We want to make this task close to the real scenario where a person records the same visual instance, *e.g.*, their dog, across multiple places and situations. Our strategy is to emulate such a scenario by finding YouTube channels that frequently mention the same instance across multiple videos. While Merlot Reserve is large and diverse, only few channels are represented with more than one video in the dataset. Instead, we download all videos and automatic transcripts from the channels of 15 popular YouTube bloggers. We then run our mining algorithm, but manu-

ally supervise the steps for visual filtering and finding additional instance samples. We first manually verify that each instance name and its visual reference are good matches. Then, we find additional sample shots across *all videos in the channel* by ranking them according to their visual similarity to the instance reference shot. Finally, we review and label the top 1000-scored shots for each named instance. The **supplementary material** includes a screenshot of the annotation tool. In total, this subset includes 15 named instances with more than 686 labeled samples. Figure 5 (bottom left) shows two example instances in our dataset.

Query-time Dataset $\mathcal{Q}$. Our end goal is to retrieve named instances via natural language queries. We would like to be able to find videos when `<my dog biscuit> is grabbing a pink Frisbee.` To this end, we manually caption 30 instance samples with descriptive queries. Thus, the Query-time dataset includes (manually captioned) video-caption pairs containing instances from the Test-time Personalization dataset. Figure 5 (bottom right) shows manually curated captions for two instances in our dataset.

5. Experiments

Our experiments first ablate our model's contributions and loss design (Section 5.1), and evaluate our final model in personalized instance retrieval benchmarks (Section 5.2).

Evaluation Datasets and Metrics. We evaluate our approach on two datasets: (i) our newly introduced *This-Is-My* personal video instance retrieval benchmark, and (ii) the personalized image retrieval benchmark built on DeepFashion2 [10] proposed by [5]. We evaluate retrieval performance in two settings:

1) Generic Instance Retrieval: In this case, we build queries for learned instances using a generic prompt, *i.e.*, "An image of *", and measure retrieval performance using mean Average Precision (mAP) and Mean Reciprocal Rank (MRR). In this setting, there are multiple correct matches.

2) Contextualized Instance Retrieval: In this case, we use natural language queries that describe the personal instance in a specific context, *e.g.*, using a scene description such as "A photo of * lying on the beach.". In this case, we assume there is only a single correct match, and we measure performance with MRR and Recall-at-5 (R@5).

Implementation Details. We use the ViT-B/16 version of CLIP in most of our experiments if not otherwise indicated. All the learnable parameters z_i and C_l are randomly initialized from $\mathcal{N}(0, 0.1)$. We set the number of category features to $q = 512$ and the number of instance tokens to $n_w = 1$ by default. No data augmentation is used during training (the visual embeddings thus have to be computed just once). For meta-personalization of category features C_l, we randomly select 32 named instances from $\mathcal{D}$ that are 0-shot classified to each category l and train for 20 epochs. This process is repeated 10 times, re-initializing z_i each time while retaining C_l from the previous run. Although updating the parameters of C_l to a small set of instances $\mathcal{P}$ at test time is beneficial, there is a risk of overfitting when only a few or just a single instance of each category is provided. To mitigate this, we include additional instances from $\mathcal{D}$ with the same categories as contained in $\mathcal{P}$ during test time personalization. All our results are averaged over five runs, each with different random seeds, and we report the standard error for each experiment.

Table 1. **Ablation Experiments.** We verify our model and training objective design through ablations on *This-Is-My* and DeepFashion2. We report personal instance retrieval performance in terms of mAP and MRR (higher is better).

Ablation	*This-Is-My* mAP	*This-Is-My* MRR	DeepFashion2 mAP	DeepFashion2 MRR
a) w/o meta-pers.	54.1±1.3	83.1±2.1	35.2±0.5	55.2±1.9
b) single C	<u>55.5</u>±0.4	<u>86.9</u>±0.5	44.1±0.5	64.7±1.5
c) w/o $\mathcal{L}_l$	53.9±1.0	86.4±2.2	**47.3**±0.8	68.3±1.2
d) w/o $\mathcal{L}_c$	48.9±1.0	78.6±2.4	<u>47.1</u>±0.5	<u>68.4</u>±0.8
e) $\mathcal{N} = \mathcal{B}$	47.4±0.4	73.7±1.8	-	-
f) w/o pre-trained C	53.0±0.7	85.1±3.0	44.3±0.8	67.1±1.0
Ours	**56.4±0.6**	**87.4±1.2**	**47.3±0.7**	**69.9±0.8**

5.1. Ablations

In Table 1 we report results in the generic instance retrieval setting for the following ablations to illustrate the effect of our model and training objective design:

a) w/o meta-personalization: In this case, we do not learn a meta-personalized C and instead directly learn instance embeddings w_i. This ablation demonstrates that learning global category attributes through meta-personalization improves generalization at test-time personalization.

b) single C (shared for all categories): In this experiment, we learn only a single category matrix C, which is shared among all the categories. The results demonstrate that learning separate attributes per category is better than sharing global attributes among all categories.

c) w/o $\mathcal{L}_l$: We explore the influence of the language-language contrastive loss $\mathcal{L}_l$. The benefits of $\mathcal{L}_l$ are more pronounced in *This-Is-My*, where instances belong to different categories.

d) w/o $\mathcal{L}_c$: We analyze the contribution of the category-anchoring loss $\mathcal{L}_c$. Including $\mathcal{L}_c$ is important on *This-Is-My*, likely because it keeps the learning focused on the category and prevents it from capturing other scene features. This is less the case on DeepFashion2, where images are very object-centric, and there is less background variety.

e) $\mathcal{N} = \mathcal{B}$ (w/o non-instance segments): We do not use non-instance segments $\mathcal{N}$ (*i.e.*, segments of the same video but low vision-language similarity) as additional negatives in this ablation. Including $\mathcal{N}$ is important, as it can help prevent the instance embedding from capturing non-instance nuisance features from the scene.

Figure 6. **Contextualized *This-Is-My* Retrievals.** We show personalized query-time retrievals for four *This-Is-My* instances. Search prompts are shown on the left and correct retrievals are highlighted in green.

f) w/o pre-trained C: In this case, we randomly initialize C instead of using a meta-personalized C. Compared to a) the sharing of category attributes among fashion items in DeepFashion2 shows clear benefits. This is less the case for *This-Is-My*, where instances belong to different categories.

5.2. Personal Instance Retrieval Benchmarks

Baselines. To demonstrate the effectiveness of our approach, we compare it to prior work [5] and the following strong baselines: **CLIP (visual)**, which represents each instance as the average visual embedding of the corresponding training examples, **CLIP (language)**, which represents each instance with a 0-shot prompt embedding of the corresponding category name, and **CLIP (V+L)**, which uses the average of the visual and language embedding.

Named Video Instance Retrieval on *This-Is-My*. We evaluate our approach on the 15 named instances in our *This-Is-My* test-time personalization dataset $\mathcal{P}$. To learn personal instance tokens, we train on video shots that occur in the video where the instance was named. All other video shots (belonging to other videos) are in the retrieval corpus for evaluation. Furthermore, we also include all other shots that do not contain the instance in the retrieval corpus as distractors. For contextualized retrieval we use the manually collected text captions from the query-time dataset $\mathcal{Q}$ as queries. We compare against the baselines and report contextualized retrieval performance on the left and generic instance retrieval performance on the right of Table 2. Our model clearly outperforms the baselines in both settings. Interestingly, the visual and language baselines have opposing advantages in generic vs. contextualized retrieval, while our model performs well in both settings. Qualitative retrieval results are shown in Figures 6.

Fashion Item Retrieval on DeepFashion2. We follow the setup of [5] and consider DeepFashion2 [10] for personalized fashion item retrieval. The dataset in this setting consists of 653 training and 221 evaluation images across 50 instances (*i.e.*, unique fashion items). While our focus is on video retrieval, our model and training objective can be used for image retrieval with minimal adjustments. We perform meta-personalization by pre-training on other non-instance-labeled images of DeepFashion2 (using only a single training image per instance). For a fair comparison with [5], the instance tokens are learned by randomly sampling $k = 5$ images per instance for training. We compare to results from [5] and the baselines in Table 3. Results using contextualized queries provided by [5] are on the left, and generic instance retrieval performance on the right. Note that [5] does not provide results for generic instance retrieval so we leave these two columns empty (-). Our approach outperforms the baselines and prior work by a large margin.

Table 2. ***This-Is-My* Video Instance Retrieval Task.** We report personal instance retrieval performance in language-specified contexts (*e.g.*, "* catching a pink frisbee") on the left and generic instance retrieval (*e.g.*, "An image of *") on the right.

	Context. Retr.		Generic Retr.	
Method	MRR	R@5	mAP	MRR
Random	0.0±0.0	0.0±0.0	1.1±0.2	2.2±1.3
CLIP (language)	30.8±0.0	36.7±0.0	16.6±0.0	44.2±0.0
CLIP (visual)	10.3±1.0	12.0±1.6	48.0±0.6	75.0±3.2
CLIP (V+L)	20.9±1.6	23.3±2.1	51.7±0.4	81.9±0.0
Ours	**42.0±1.3**	**50.7±1.8**	**56.4±0.6**	**87.4±1.2**

Table 3. **Fashion Item Retrieval on DeepFashion2.** We evaluate our approach on the personalized instance retrieval task with contextualized queries from [5] (left) and generic instance retrieval (right). Results with * use ViT-B/32 instead of ViT-B/16.

	Context. Retr.		Generic Retr.	
Method	MRR	R@5	mAP	MRR
Random	2.9±0.2	1.8±0.5	4.7±0.5	9.5±2.0
CLIP (language)	21.2±0.0	25.3±0.0	8.3±0.0	16.9±0.0
CLIP (visual)	14.2±0.3	17.3±0.3	20.6±0.4	43.7±1.1
CLIP (V+L)	20.8±0.8	25.4±1.7	20.5±0.5	42.8±1.2
Adapter* [5]	5.9±0.7	-	-	-
COLLIE* [5, 29]	7.9±0.7	-	-	-
PALAVRA* [5]	28.4±0.7	39.2±1.3	-	-
Ours*	34.4±0.7	45.2±1.1	40.0±1.0	69.3±1.8
Ours	**38.4±0.4**	**51.4±0.4**	**53.4±0.4**	**77.7±0.6**

6. Conclusion

We have introduced a meta-personalization approach for learning to retrieve named instances in video given a natural language query. We demonstrated the effectiveness of our approach on two datasets and showed that it outperforms strong baselines. Our effort is a step towards vision-language models trained over a large number of general and personal concepts. Our approach opens up the possibility of personalized language-based video editing, summarization, and generation, *e.g.*, identify the key instances in a collection of footage and create an edited story about the instances using natural language.

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, Andy 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. *ArXiv*, abs/2204.14198, 2022. 1

[2] Fernando Amat, Ashok Chandrashekar, Tony Jebara, and Justin Basilico. Artwork personalization at netflix. In *Proceedings of the 12th ACM conference on recommender systems*, pages 487–488, 2018. 2

[3] Soulef Benhamdi, Abdesselam Babouri, and Raja Chiky. Personalized recommender system for e-learning environment. *Education and Information Technologies*, 22(4):1455–1477, 2017. 2

[4] Arantxa Casanova, Marlene Careil, Jakob Verbeek, Michal Drozdzal, and Adriana Romero Soriano. Instance-conditioned gan. *Advances in Neural Information Processing Systems*, 34:27517–27529, 2021. 2

[5] Niv Cohen, Rinon Gal, Eli A Meirom, Gal Chechik, and Yuval Atzmon. "This is my unicorn, Fluffy": Personalizing frozen vision-language representations. In *ECCV*, 2022. 2, 7, 8

[6] Chelsea Finn, P. Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017. 3

[7] Chelsea Finn, Aravind Rajeswaran, Sham Kakade, and Sergey Levine. Online meta-learning. In *International Conference on Machine Learning*, pages 1920–1930. PMLR, 2019. 3

[8] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. *ArXiv*, abs/2208.01618, 2022. 2

[9] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *arXiv preprint arXiv:2110.04544*, 2021. 2

[10] Yuying Ge, Ruimao Zhang, Xiaogang Wang, Xiaoou Tang, and Ping Luo. Deepfashion2: A versatile benchmark for detection, pose estimation, segmentation and re-identification of clothing images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5337–5345, 2019. 7, 8

[11] Satya Krishna Gorti, Noël Vouitsis, Junwei Ma, Keyvan Golestan, Maksims Volkovs, Animesh Garg, and Guangwei Yu. X-pool: Cross-modal language-video attention for text-video retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5006–5015, 2022. 2

[12] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey A. Gritsenko, Diederik P. Kingma, Ben Poole, Mohammad Norouzi, David J. Fleet, and Tim Salimans. Imagen video: High definition video generation with diffusion models. *ArXiv*, abs/2210.02303, 2022. 1

[13] De-An Huang, S. Buch, Lucio M. Dery, Animesh Garg, Li Fei-Fei, and Juan Carlos Niebles. Finding "it": Weakly-supervised reference-aware visual grounding in instructional videos. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5948–5957, 2018. 2

[14] 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 *International Conference on Machine Learning*, pages 4904–4916. PMLR, 2021. 1, 2

[15] Yihan Jiang, Jakub Konečnỳ, Keith Rush, and Sreeram Kannan. Improving federated learning personalization via model agnostic meta learning. *arXiv preprint arXiv:1909.12488*, 2019. 2

[16] Alexander Kunitsyn, Maksim Kalashnikov, Maksim Dzabraev, and Andrei Ivaniuta. Mdmmt-2: Multidomain multimodal transformer for video retrieval, one more step towards generalization. *arXiv preprint arXiv:2203.07086*, 2022. 2

[17] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7331–7341, 2021. 2

[18] Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in neural information processing systems*, 34:9694–9705, 2021. 1, 2

[19] Shuang Li, Yilun Du, Joshua B Tenenbaum, Antonio Torralba, and Igor Mordatch. Composing ensembles of pre-trained models via iterative consensus. *arXiv preprint arXiv:2210.11522*, 2022. 1, 2

[20] Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. Clip4clip: An empirical study of clip for end to end video clip retrieval and captioning. *Neurocomputing*, 508:293–304, 2022. 2

[21] Yiwei Ma, Guohai Xu, Xiaoshuai Sun, Ming Yan, Ji Zhang, and Rongrong Ji. X-clip: End-to-end multi-grained contrastive learning for video-text retrieval. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 638–647, 2022. 2

[22] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 1, 2, 4, 5

[23] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. Technical report, Tech. Rep., Technical report, OpenAI, 2022. 4

[24] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *ArXiv*, abs/2204.06125, 2022. 1

[25] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. *ArXiv*, abs/2102.12092, 2021. 1

[26] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. *arXiv preprint arXiv:2208.12242*, 2022. 2

[27] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L. Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, Seyedeh Sara Mahdavi, Raphael Gontijo Lopes, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. *ArXiv*, abs/2205.11487, 2022. 1

[28] Uriel Singer, Adam Polyak, Thomas Hayes, Xiaoyue Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman. Make-a-video: Text-to-video generation without text-video data. *ArXiv*, abs/2209.14792, 2022. 1

[29] Gabriel Skantze and Bram Willemsen. Collie: Continual learning of language grounding from language-image embeddings. *Journal of Artificial Intelligence Research*, 74:1201–1223, 2022. 8

[30] Tomáš Souček and Jakub Lokoč. Transnet v2: An effective deep network architecture for fast shot transition detection. *arXiv preprint arXiv:2008.04838*, 2020. 4

[31] Yu Sun, X. Wang, Zhuang Liu, John Miller, Alexei A. Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *ICML*, 2020. 3

[32] Ruben Villegas, Mohammad Babaeizadeh, Pieter-Jan Kindermans, Hernan Moraldo, Han Zhang, Mohammad Taghi Saffar, Santiago Castro, Julius Kunze, and D. Erhan. Phenaki: Variable length video generation from open domain textual description. *ArXiv*, abs/2210.02399, 2022. 1

[33] Changhan Wang, Yun Tang, Xutai Ma, Anne Wu, Dmytro Okhonko, and Juan Miguel Pino. Fairseq s2t: Fast speech-to-text modeling with fairseq. In *AACL*, 2020. 4

[34] Dequan Wang, Shaoteng Liu, Sayna Ebrahimi, Evan Shelhamer, and Trevor Darrell. On-target adaptation. *ArXiv*, abs/2109.01087, 2021. 3

[35] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno A. Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *ICLR*, 2021. 3

[36] Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Dualprompt: Complementary prompting for rehearsal-free continual learning. In *ECCV*, 2022. 2

[37] Sangmin Woo, Jinyoung Park, Inyong Koo, Sumin Lee, Minki Jeong, and Changick Kim. Explore and match: End-to-end video grounding with transformer. *arXiv preprint arXiv:2201.10168*, 2022. 2

[38] Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, and Ludwig Schmidt. Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7959–7971, 2022. 2

[39] 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. *arXiv preprint arXiv:2109.14084*, 2021. 2

[40] 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. *arXiv preprint arXiv:2209.06430*, 2022. 2

[41] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Tubedetr: Spatio-temporal video grounding with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16442–16453, 2022. 2

[42] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Zero-shot video question answering via frozen bidirectional language models. *arXiv preprint arXiv:2206.08155*, 2022. 1, 2

[43] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pages 1094–1100. PMLR, 2020. 3

[44] Rowan Zellers, Jiasen Lu, Ximing Lu, Youngjae Yu, Yanpeng Zhao, Mohammadreza Salehi, Aditya Kusupati, Jack Hessel, Ali Farhadi, and Yejin Choi. Merlot reserve: Neural script knowledge through vision and language and sound. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16375–16387, 2022. 2, 6