# Every Shot Counts: Using Exemplars for Repetition Counting in Videos

Saptarshi Sinha[1] [0000−0002−5207−1551], Alexandros Stergiou[2] [0000−0003−4706−4231], and Dima Damen[1] [0000−0001−8804−6238]

[1] University of Bristol, UK [2] University of Twente, NL
https://sinhasaptarshi.github.io/escounts

**Abstract.** Video repetition counting infers the number of repetitions of recurring actions or motion within a video. We propose an exemplar-based approach that discovers visual correspondence of video exemplars across repetitions within target videos. Our proposed **E**very **S**hot **Counts** (ESCounts) model is an attention-based encoder-decoder that encodes videos of varying lengths alongside exemplars from the same and different videos. In training, ESCounts regresses locations of high correspondence to the exemplars within the video. In tandem, our method learns a latent that encodes representations of general repetitive motions, which we use for exemplar-free, zero-shot inference. Extensive experiments over commonly used datasets (RepCount, Countix, and UCFRep) showcase ESCounts obtaining state-of-the-art performance across all three datasets. Detailed ablations further demonstrate the effectiveness of our method.

**Keywords:** Video Repetition Counting · Video Exemplar · Cross-Attention Transformer · Video Understanding

## 1 Introduction

In recent years, tremendous progress has been made in video understanding. Visual Language Models (VLMs) have been adopted for many vision tasks including video summarisation [30,43,58], localisation [44,55], and question answering (VQA) [1,19,38,62]. Despite their great success, recent analysis [21] shows that VLMs can still fail to count objects or actions correctly. Robust counting can be challenging due to appearance diversity, limited training data, and the semantic ambiguity of identifying 'what' to count.

Evidence in developmental psychology and cognitive neuroscience [49,52,53] shows that infants fail to differentiate the number of hidden objects if not shown and counted to them first, suggesting an upper limit of individual objects in working memory. However, infants exposed to an instance of the object first could better approximate carnality. This shows that counting is a visual exercise of matching to exemplars, and is developed before understanding their semantics.

Object-counting in images has recently exploited exemplars to improve performance [32,37]. In training, models attend to one or more exemplars of 'what'

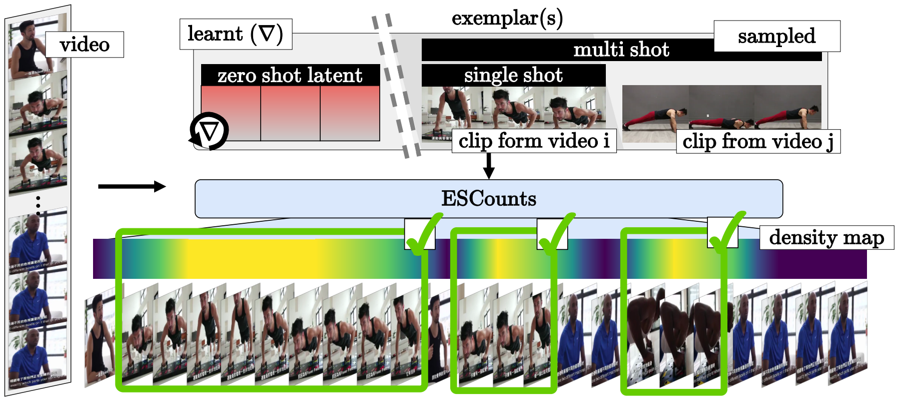


**Fig. 1: VRC with ESCounts** involves exemplars for relating information of the repeating action across the video. We visualise the density map with high relevance regions to the action *push-up* being highlighted, whilst regions of low relevance are not.

object(s) to count alongside learnt embeddings for exemplar-free counting. During inference, only the learnt embeddings are used for zero-shot counting without knowledge from exemplars. As videos are of variable length and repetition durations vary, these approaches are not directly applicable to videos.

Taking inspiration from image-based approaches, we address Video Repetition Counting (VRC) with exemplars for the first time. We differ from prior works that formulate VRC as classifying a preset number of repetitions [6,14,65], or detecting relevant parts (start/end) of repetitions [13, 18, 27]. Instead, we argue that learning correspondences to reference exemplar(s) during training can provide a strong prior for discovering correspondences across repetitions at inference. We propose **E**very-**S**hot **Counts** (ESCounts), a transformer-based encoder-decoder that during training encodes videos of varying lengths alongside exemplars and learns latents of general repeating motions, as shown in Fig. 1. Similar to [18, 27], we use density maps to regress the temporal location of repetitions. At inference, learnt latents are used for exemplar-free counting.

In summary, our contributions are as follows: (i) We introduce exemplar-based counting for VRC (ii) We propose an attention-based encoder-decoder that corresponds exemplars to a query video of varying length. (iii) We learn latents for general repeating features and use them to predict the number of repetitions during inference without exemplars, (iv) We evaluate our approach on the three commonly-used VRC datasets: RepNet [18], Countix [14], and UCFRep [63]. Our approach achieves a new state-of-the-art in every benchmark, even on Countix where start-end times of repetitions are not annotated.

## 2 Related Works

We first review methods for the long-established task of object counting in images. We then review VRC methods for videos.

### 2.1 Object Counting in Images

Methods can be divided into class-specific and class-agnostic object counting.
**Class-specific counting**. These methods learn to count objects of singular classes or sets of categories *e.g.* people [26,54], cars [17], or wildlife [4]. A large portion of object-counting approaches [10,17,39,46,54] have relied on detecting target objects and counting their instances. Traditional methods have used hand-crafted feature descriptors to detect human heads [54] or head-shoulders [26] for crowd-counting. Other methods have used blobs [23], individual points [33], and object masks [11] for detecting and counting instances. Though object detection can be a preliminary step before counting, detection methods rely strongly on the object detector's performance which can be less effective in densely crowded images [10]. Other methods instead relied on regression, to either regress to the target count [10,59] or estimate a density map [24,40,64].
**Class-agnostic counting**. Class-specific counting approaches are impractical for general settings where prior knowledge of the object category is not available. Recent works [3,29,32,47] have used one (or a few) exemplars as references to estimate a density map for unknown target classes. Building on the property of *image self-similarity*, [37] proposed a convolutional matching network. They cast counting as an image-matching problem, where exemplar patches from the same image are used to match against other patches within the image. Following up, Liu *et al.* [32] used an encoder for the query image, a convolution-based encoder for the exemplar, and an interaction module to cross-attend information between the exemplar and the image. A convolutional decoder was used to regress the density map. Recent approaches have also fused text and visual embeddings [3], used contrastive learning across modalities [21], and generated exemplar prototypes using stable diffusion [60]. Inspired by these methods, we propose an attention-based encoder-decoder that extends exemplar-based counting to VRC. Our approach is invariant to video lengths and can use both learnt or encoded exemplars.

### 2.2 Video Repetition Counting (VRC)

Compared to image-based counting, video repetition counting has been less explored. Early approaches have compressed motion into a one-dimensional signal and recovered the repetition structure from the signal's period [2, 22, 36, 41]. The periodicity can then be counted with Fourier analysis [2, 5, 7, 12, 22, 42], peak detection [51], or wavelet analysis [45]. However, these methods are limited to uniformly periodic repetitions. For non-periodic repetitions, temporal understanding frameworks [9,31,35,48] have been adapted. Zhang *et al.* [63] proposed a context-aware scale-insensitive framework to count repetitions of varying scales and duration. Their method exhaustively searches for pairs of consecutive repetitions followed by a prediction refinement module. Recent methods [13,14,18] have also extended image self-similarity to the temporal dimension with Temporal Self-similarity Matrices (TSM). TSM is constructed using pair-wise similarity of embeddings over temporal locations. RepNet [14] used a transformer-based period predictor. To count repetitions with varying speeds, Trans-RAC [18] modified TSM to use multi-scale sequence embeddings. For counting under poor lighting conditions, [65] used both audio and video in a multi-modal framework. They selectively aggregated information from the two modalities using a reliability estimation module. Li *et al*. [27] also used multi-modal inputs with optical flow as an additional signal supporting RGB for detecting periodicity.

Recent works attempt to utilise spatial [61] or temporal [28, 66] saliency for repetition counting. Yao *et al*. [61] proposed a lightweight pose-based transformer model that used action-specific salient poses as anchors. The need for salient pose labels for each action limits generalisability to unseen repetitions. Zhao *et al*. [66] used a dual-branch architecture to first select repetition-relevant video segments and then attend over these frames. Li *et al*. [28] used a joint objective to localise and binary classify regions as (non-)repetitive.

The above methods do not utilise the correspondences discovered by exemplar repetitions. Thus, do not relate variations in the action's performance. We propose using action exemplars as references for VRC. Exemplars have previously been used in videos for action recognition tasks [16, 20, 56, 57]. [56] used silhouette/pose exemplars for classifying action sequences into predefined categories. [57] converted training videos to a visual vocabulary and used the most discriminative visual words as exemplars. These methods are limited to a predefined set of classes. To our knowledge, we are the first to use exemplars for repetition counting in videos.

## 3 Every Shot Counts (ESCounts) Model

In this section, we introduce our ESCounts model (overviewed in Fig. 2). We formally define encoding variable length videos alongside our model's output, in Sec. 3.1. We introduce the attention-based decoder that corresponds the input video to training exemplars and learnt latents in Sec. 3.2. Predictions over temporally shifted inputs are then combined, detailed in Sec. 3.3.

### 3.1 Input Encoding and Output Prediction

We denote the full **video** as $\mathbf{v}$ of varying $\mathcal{T}$ length and fixed $H \times W$ spatial resolution. Segment $\mathbf{e}_s$ containing a single instance of the repeating action we wish to count, is selected as an **exemplar**. Exemplars are defined based on provided [start, end] labels of every repetition in the video[1]. During training, we select one or more exemplar shots $\mathcal{S} \subseteq \{\mathbf{e}_1, \dots, \mathbf{e}_s\}$. Each training instance is a combination of the query video and the set of exemplars $(\mathbf{v}, \mathcal{S})$.

We tokenise and encode the video $\mathbf{v}$ from its original size $\mathcal{T} \times H \times W$ into spatiotemporal latents $\mathbf{z}_v$. To account for the video's variable length, encoder $\mathcal{E}$ is applied over a fixed-size sliding window. The encoded video is represented

[1] For datasets where the start/end times are not available, pseudo-labels are used instead by uniformly dividing the video by the ground truth count.

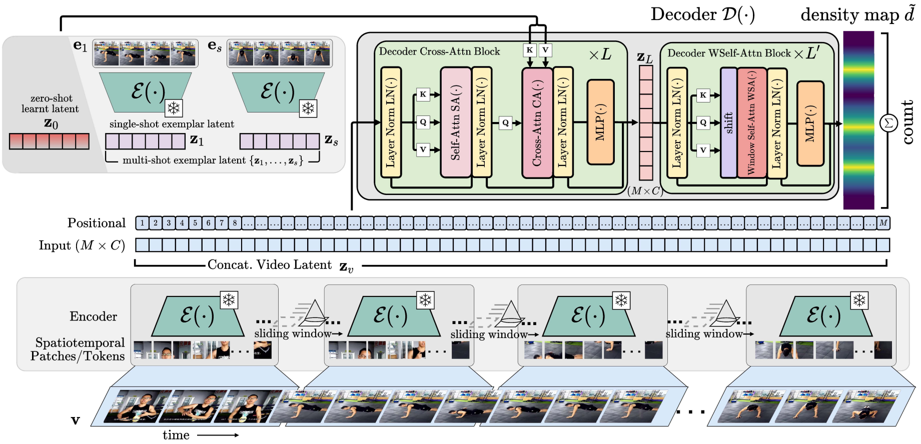


**Fig. 2: ESCounts Model overview**. **Bottom:** Video $\mathbf{v}$ is encoded by $\mathcal{E}$ over sliding temporal windows to spatiotemporal latents $\mathbf{z}_v \in \mathbb{R}^{M \times C}$. **Top Left:** Exemplars $\{\mathbf{e}_s\}$ are also encoded with $\mathcal{E}$. **Top Right:** Video $\mathbf{z}_v$ and exemplar $\mathbf{z}_s$ latents are cross-attended by decoder $\mathcal{D}$ over $L$ cross-attention blocks. The resulting $\mathbf{z}_L \in \mathbb{R}^{M \times C}$ are attended over $L'$ window self-attention blocks and projected into density map $\tilde{\mathbf{d}}$. The decoder $\mathcal{D}$ is trained to regress the error between predicted $\tilde{\mathbf{d}}$ and ground truth $\mathbf{d}$ density maps. At inference, the count is obtained by summing $\tilde{\mathbf{d}}$.

by $\mathbf{z}_v \in \mathbb{R}^{M \times C}$ of $M = \mathcal{T}'H'W'$ spatiotemporal resolution with $C$ channels. We note that $M$ is not a fixed number, as it depends on the video's length $\mathcal{T}$. We add sinusoidal positional encoding to account for the relative order of these spatiotemporal latents while accommodating the variable video length.

For training only, we select exemplars $\mathcal{S}$ from either the same video or another video of the same action category; e.g. given a video containing *push-up* actions, we can sample exemplars from other videos showcasing the same action within the training set. We define a probability $p$ of sampling the exemplar from a different video; i.e. $p = 0$ implies exemplars are only sampled from the same video, whereas for $p = 1$ exemplars are always sampled from another video[2]. We sample exemplars randomly from the labelled repetitions of the video. We use $\mathcal{E}$ to encode latent representations from each exemplar $\mathbf{e}_s \in \mathcal{S}$. We use the same encoder $\mathcal{E}$ for encoding $\mathbf{v}$ and $\mathbf{e}_s$ to enable direct correspondence.

We construct the ground truth **density map d** from the labelled repetitions in the video as a 1-dimensional vector. To match the downsampled temporal resolution of our input video $\mathcal{T}'$, we also temporally downsample the ground-truth labels. The density map takes low values ($\approx 0$) at temporal locations without repetitions and high values within repetitions. We use a normal distribution $\mathcal{N}$ centred around each repetition with $(\mu_i = \frac{t_s + t_e}{2}, \sigma)$, where $t_e$ and $t_s$ are the start and end times of each repetition $i$.

$$\mathbf{d}_t = \sum_i \mathcal{N}(t; \mu_i, \sigma) \quad \forall\, t \in \{1, \ldots, \mathcal{T}'\} \tag{1}$$

[2] We ablate $p$ in our experiments.

Note that the sum of the density map $\mathbf{d}$ matches the ground truth count, i.e. $\sum \mathbf{d} = c$ where $c$ is the ground truth count for the video.

### 3.2 Latent Exemplar Correspondence

Given both the encoded video $\mathbf{z}_v = \mathcal{E}(\mathbf{v})$ and exemplars $\mathbf{z}_s = \mathcal{E}(\mathbf{e}_s) \ \forall \ \mathbf{e}_s \in \mathcal{S}$, we use an attention-based decoder $\mathcal{D}(\mathbf{z}_v, \mathbf{z}_s)$ to learn a correspondence between every repetition in the video $v$ and the encoded exemplar. Decoder $\mathcal{D}$ takes the encoded video $\mathbf{z}_v$ as input and predicts the location of every repetition in the video. The decoder outputs a 1-dimensional predicted density map of length $\mathcal{T}'$ corresponding to the occurrences of the repeating action given the exemplars.

**Cross-attention Blocks**. We explore the similarity between exemplars and query video representations to predict the corresponding locations of repetitions that match the exemplar. Thus, inspired by [32], we use cross-attention to relate exemplar and video encodings. We define $L$ cross-attention blocks. Each block initially Self-Attends SA$(\cdot)$ the video latents $\mathbf{z}_l \in \mathbb{R}^{M \times C}$ with multi-head self-attention.

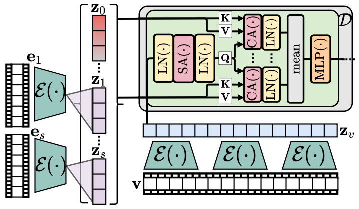


**Fig. 3: Cross-Attention block**. Video latents $\mathbf{z}_v$ are self-attended and then cross-attended with latents $\mathbf{z}_s$ from each exemplar $s \in \mathcal{S}$ and the learnt latent $\mathbf{z}_0$ with the same weights. The resulting representations are then averaged.

We note that for the first layer, $\mathbf{z}_1 = \mathbf{z}_v$. We then relate exemplar and video by Cross-Attending CA$(\cdot)$ video and exemplar encodings. The block's initial self-attention operation is formulated as:

$$\mathbf{z}'_l = \text{SA}(\text{LN}(\mathbf{z}_l)) + \mathbf{z}_l \ \ \forall \, l \in \{1, \ldots, L\}, \tag{2}$$

where LN$(\cdot)$ is Layer Normalisation. It is essential to self-attend across the video first to capture the features of the repeated actions within the video, and enforce feature correspondence between repetitions.

Repetitions can vary by viewing angles, performance, or duration. We thus wish to allow a varying number of exemplars for counting a repeating action, as shown in Fig. 3. Given a selected number of exemplar shots $\mathcal{S}$, we apply CA in parallel with $\mathbf{z}'_l$ used as a shared query $\mathbf{Q}$ and each of the $\mathcal{S}$ exemplars used as keys and values $\mathbf{K}, \mathbf{V}$ enabling the fusion of repetition-relevant information. As the latents of the video are used as queries $\mathbf{Q}$, spatiotemporal resolution $M$ is maintained. Outputs are then averaged:

$$\mathbf{z}''_l = \frac{1}{N} \sum_{s=1}^{\mathcal{S}} \text{CA}(\mathbf{z}_s, \text{LN}(\mathbf{z}'_l)) + \mathbf{z}'_l \ \ \forall \, l \in \{1, \ldots, L\}, \tag{3}$$

where $\mathcal{S}$ is the set of exemplars selected and $\mathbf{z}_s$ is the latent for the $s^{\text{th}}$ exemplar.

We also want to learn repeating motions to estimate repetitions without explicitly providing exemplars. We thus define a learnable latent $\mathbf{z}_0$ to cross-attend

$\mathbf{z}_v$. At each training step, we select exemplars from $\{\mathbf{z}_0, \mathbf{z}_1, ..., \mathbf{z}_s\}$ and perform CA with $\mathbf{z}_0$ or $\{\mathbf{z}_1, ..., \mathbf{z}_s\}$. **Importantly, at inference, we use only $\mathbf{z}_0$.**

We obtain the cross-attention blocks' output, defined as $\mathbf{z}_{l+1} \in \mathbb{R}^{M \times C}$, with a Multi-Layer Perceptron MLP on the exemplar-fused latents $\mathbf{z}''_l$.

$$\mathbf{z}_{l+1} = \text{MLP}(\text{LN}(\mathbf{z}''_l)) + \mathbf{z}''_l \;\; \forall\, l \in \{1, \dots, L\} \tag{4}$$

**Window Self-attention Blocks**. We explore the spatio-temporal inductive bias within the self-attention blocks. For this, each latent attends locally to its spatio-temporal neighbouring tokens, over $L'$ Window Self-Attention WSA$(\cdot)$ [34] layers. We denote $\forall\, l \in \{L+1, \dots, L+L'\}$ :

$$\mathbf{z}_{l+1} = \text{MLP}(\text{LN}(\mathbf{z}'_l)) + \mathbf{z}'_l, \text{ where } \mathbf{z}'_l = \begin{cases} \text{WSA}(\text{LN}(\mathbf{z}_l)) + \mathbf{z}_l, \text{ if } l = L+1 \\ \text{WSA}(\text{shift}(\text{LN}(\mathbf{z}_l))) + \mathbf{z}_l, \text{ else} \end{cases} \tag{5}$$

where WSA is window self-attention. Note that following [34] windows are shifted at each layer to account for connections across different windows.

The output of the WSA blocks is of size $\mathbf{z}_{L+L'} \in \mathbb{R}^{M \times C}$ . In turn, $\mathbf{z}_{L+L'}$ encodes repetition-relevant features over space and time and is used to predict density map $\tilde{\mathbf{d}}$ for the occurrences of the target repeating action over time. We use a fully connected layer to project the latent to a 1-channel vector, i.e. MLP: $\mathbb{R}^{M \times C} \rightarrow \mathbb{R}^{M}$. We then vectorise the spatial resolution $H'W'$ whilst maintaining $\mathcal{T}'$ resulting to the predicted density map $\tilde{\mathbf{d}} \in \mathbb{R}^{\mathcal{T}'}$.

**Training Objective**. Given ground-truth $\mathbf{d}$ and the predicted $\tilde{\mathbf{d}} = \mathcal{D}(\mathbf{z}_v, \mathbf{z}_{\underline{s}})$ density maps, we train $\mathcal{D}$ to regress the *Mean Square Error* between $\mathbf{d}$ and $\tilde{\mathbf{d}}$, and following [65], the *Mean Absolute Error* between ground truth counts $c$ and the predicted counts $\tilde{c}$ obtained by linearly summing the density map $\tilde{c} = \sum \tilde{\mathbf{d}}$

$$\mathcal{L} = \underbrace{\frac{||\mathbf{d} - \tilde{\mathbf{d}}||^2}{\mathcal{T}'}}_{\text{MSE}(\mathbf{d}, \tilde{\mathbf{d}})} + \underbrace{\frac{|c - \sum \tilde{\mathbf{d}}|}{c}}_{\text{MAE}(c, \tilde{c})} \tag{6}$$

At inference, we use the predicted count $\tilde{c}$.

### 3.3 Time-Shift Augmentations

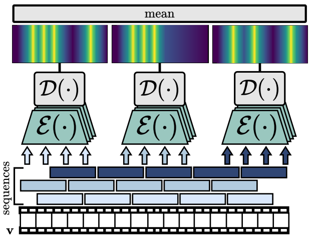


**Fig. 4: Shifted Density maps** from each video, are meaned to $\tilde{\mathbf{d}}$.

The predicted density map $\tilde{\mathbf{d}}$ results from encoding a video with $\mathcal{E}$, over non-overlapping sliding windows. However, as each window is of fixed temporal resolution, repetitions may span over multiple windows. Thus, we include time-shifting augmentations in which the start time of the encoded video is adjusted to allow for different spatiotemporal tokens. We train with augmentations of the start time whilst at inference, we use an ensemble of time-shift augmentations for a more robust estimation. We use multiple overlapping sequences as shown in Fig. 4 and combine the predicted density maps over $\mathbf{K}$ shifted start/end positions. We obtain the final predicted density map by temporally aligning and averaging the predictions; $\tilde{\mathbf{d}}_t = \frac{1}{|\mathbf{K}|} \sum_{k \in \mathbf{K}} \tilde{\mathbf{d}}^k_{t+\epsilon_k}$, where $\epsilon_k$ is the shifting for each $k \in \mathbf{K}$.

## 4 Experiments

We overview the used datasets, implementation details, and evaluation metrics in Sec. 4.1. We include quantitative and qualitative comparisons to state-of-the-art methods in Sec. 4.2. We ablate over different ESCounts settings in Sec. 4.3. For all results, we only report zero-shot counting during inference. In Sec. 4.4, we evaluate ESCounts' when exemplars are available during inference.

### 4.1 Experimental Setup

**Datasets**. We evaluate our method on a diverse set of VRC datasets.

*RepCount* [18] contains videos of workout activities with varying repetition durations. Annotations include counts alongside start and end times per repetition. We use the publicly available Part-A with 758, 131, and 152 videos for train, val, and test respectively. Additionally, we use the provided open set split with 70% categories for train, 10% for val, and 20% for testing. We tune the hyperparameters on the *val* set and report our results on the *test* set.

*Countix* [14] is a subset of Kinetics [8] containing videos of repetitive actions with $4,588$, $1,450$, and $2,719$ videos for train, val, and test respectively. Counts are provided without individual repetition start-end times. Countix does not have many pauses or interruptions between counts. Thus, we define pseudo-repetition annotations by dividing videos into uniform segments based on the ground truth count. The pseudo-labels are used to estimate the density maps without additional annotations, to compare directly to other methods.

*UCFRep* [63] is a subset of UCF-101 [50] consisting of 420 train and 106 val videos from 23 categories with counts and annotations of start and end times. Following [27,63], we report our results on the *val* split as no *test* set is available.

**Implementation Details**. Unless specified otherwise, we use MAE-pretrained ViT-B [15] as our encoder $\mathcal{E}$ with Kinetics-400 [8] weights. We sample frames from variable-length videos every 4 frames using a sliding window of 64 frames. At each window, our encoder's input is of $16 \times 224 \times 224$ size, and the output is $8 \times 14 \times 14$, resulting in 1568 spatiotemporal tokens. We use $C = 512$ channels[3]. The input to the decoder is of variable length $M = 1568\frac{\mathcal{R}}{64}$, where $\mathcal{R}$ is the total number of frames in the video at raw framerate. Exemplars are sampled uniformly with 16 frames between the start and end of a repetition.

The encoder is frozen and we only train the decoder and zero-shot latent $\mathbf{z}_0$. We use $L = 2$ and $L' = 3$ ablating this choice in Sec. 7 of the appendix.

[3] when encoders have a different output, we add a fully connected layer to map to $C$

We train for 300 epochs on a single Tesla V100 with a batch size of 1, to deal with variable-length videos, accumulating gradients over 8 batches. We use $5e^{-2}$ weight decay and a learning rate of $5e^{-5}$ with decay by 0.8 every 60 epochs. Per training instance, we randomly set the number of exemplars $|\mathcal{S}| \sim \{0, 1, 2\}$ and sample $\mathcal{S}$ exemplars. We set the chance of sampling exemplars from a different video to $p = 0.4$.

Only the learnt latent are used at inference to predict repetition counts. We aggregate predictions over $|\mathbf{K}| = 4$ sequences.

**Evaluation Metrics**. Following previous VRC works [14, 18, 65], we use Mean Absolute Error (MAE) and Off-By-One accuracy (OBO) as evaluation metrics, calculated as Eqs. (7) and (8) respectively. Inspired by image counting methods [3, 32], we introduce Root-Mean-Square-Error (RMSE) in Eq. (9) for VRC providing a more robust metric for diverse counts compared to MAE's bias towards small counts. We also report the off-by-zero accuracy (OBZ) in Eq. (10) as a tighter metric than the corresponding OBO for precise counts.

$$MAE = \frac{1}{|\Omega|}\sum_{i\in\Omega}\frac{|c_i - \tilde{c}_i|}{c_i}, \quad (7) \qquad OBO = \frac{1}{|\Omega|}\sum_{i\in\Omega}\mathbb{1}(|c_i - \tilde{c}_i| \leq 1), \quad (8)$$

$$RMSE = \sqrt{\frac{1}{|\Omega|}\sum_{i\in\Omega}(c_i - \tilde{c}_i)^2}, \quad (9) \qquad OBZ = \frac{1}{|\Omega|}\sum_{i\in\Omega}\mathbb{1}(|c_i - \tilde{c}_i| = 0), \quad (10)$$

where $c_i$, $\tilde{c}_i$ are the ground-truth and predicted counts for $i$-th video in test set $\Omega$. $\mathbb{1}$ is the indicator function.

### 4.2 Comparison with State-of-the-art

In Tab. 1, we compare ESCounts, to prior methods on the three datasets. We provide results on the same backbone as the best-performing method on each dataset, for fair and direct comparison to previous works.

**RepCount**. Tab. 1a shows that ESCounts outperforms recent methods [18, 27, 28, 66]. Compared to the baseline [18], we improve OBZ by +0.16 and reduce RMSE by −4.68. We test on two backbones - SwinT [34] used in [18, 27] and ViT-B used in [28]. On the same SwinT backbone, our approach outperforms [27], which uses optical flow and video in tandem, by margins of −0.09 MAE and +0.02 OBO, showcasing ESCounts' ability to learn repeating motions implicitly. With a ViT-B backbone, we outperform [28] by −0.05 MAE and +0.02 OBO.

We additionally compare ESCounts on the open set setting of RepCount-A, with non-overlapping action categories between train and test sets. ESCounts outperforms [18] significantly with −0.19 MAE and +0.32 OBO. Note that recent works do not report on this more challenging setup.

**Countix**. Compared to the state-of-the-art [14, 65] in Tab. 1b, our ESCounts consistently outperforms other models with the same R(2+1)D18 encoder. Our video-only model surpasses the audio-visual model in [65] by +0.19 OBO. Further improvements on the RMSE, MAE, and OBZ are observed with ViT-B.

**Table 1: Comparison of VRC methods**. † represents multi-modal models that use added audio or flow. * denotes results reproduced using provided checkpoints. $^*$ denotes inhouse re-training using published codes. Grayed rows in (c) represent methods that finetune the encoder. Top performances for each metric and dataset are in **bold**.

**(a)** RepCount

| | | benchmark | | | | open set | |
|---|---|---|---|---|---|---|---|
| Method | Encoder | RMSE↓ | MAE↓ | OBZ↑ | OBO↑ | MAE↓ | OBO↑ |
| RepNet [14] | R2D50 | - | 0.995 | - | 0.013 | - | - |
| TransRAC [18] | SwinT | 9.130* | 0.443 | 0.085* | 0.291 | 0.625 | 0.204 |
| MFL [27]† | SwinT | - | 0.384 | - | 0.386 | - | - |
| DeTRC [28] | ViT-B | - | 0.262 | - | 0.543 | - | - |
| SkimFocus [66] | SwinB | - | 0.249 | - | 0.517 | - | - |
| ESCounts | SwinT | 6.905 | 0.298 | 0.183 | 0.403 | - | - |
| ESCounts | ViT-B | **4.455** | **0.213** | **0.245** | **0.563** | **0.436** | **0.519** |

**(b)** Countix

| Method | Encoder | RMSE↓ | MAE↓ | OBZ↑ | OBO↑ |
|---|---|---|---|---|---|
| RepNet [14] | R2D50 | - | 0.364 | - | 0.697 |
| Sight & Sound [65]† | R(2+1)D18 | - | 0.307 | - | 0.511 |
| ESCounts | R(2+1)D18 | 3.536 | 0.293 | 0.286 | **0.701** |
| ESCounts | ViT-B | **3.029** | **0.276** | **0.319** | 0.673 |

**(c)** UCFRep

| Method | Encoder | RMSE↓ | MAE↓ | OBZ↑ | OBO↑ |
|---|---|---|---|---|---|
| Levy & Wolf [25] | RX3D101 | - | 0.286 | - | 0.680 |
| RepNet [14] | R2D50 | - | 0.998 | - | 0.009 |
| Context (F) [63] | RX3D101 | 5.761$^*$ | 0.653$^*$ | 0.143$^*$ | 0.372$^*$ |
| TransRAC [18] | SwinT | - | 0.640 | - | 0.324 |
| MFL [27]† | RX3D101 | - | 0.388 | - | 0.510 |
| ESCounts | RX3D101 | 2.004 | 0.247 | 0.343 | 0.731 |
| ESCounts | ViT-B | **1.972** | 0.216 | 0.381 | 0.704 |
| Context [63] | RX3D101 | 2.165* | 0.147 | **0.452*** | 0.790 |
| Sight & Sound [65]† | R(2+1)D18 | - | **0.143** | - | **0.800** |

**Table 2: Cross-dataset generalisation**. Arrows denote train → test datasets. Results with provided checkpoints are denoted with *.

| | RepCount → Countix | | | | RepCount → UCFRep | | | |
|---|---|---|---|---|---|---|---|---|
| | RMSE↓ | MAE↓ | OBZ↑ | OBO↑ | RMSE↓ | MAE↓ | OBZ↑ | OBO↑ |
| RepNet [14] | - | - | - | - | - | 0.998 | - | 0.009 |
| TransRAC [18] | 6.867* | 0.593* | 0.132* | 0.364* | 6.701* | 0.640 | 0.087* | 0.324 |
| MFL [27] | - | - | - | - | - | 0.523 | - | 0.350 |
| SkimFocus [66] | - | - | - | - | - | 0.502 | - | 0.391 |
| DeTRC [28] | - | - | - | - | - | 0.543 | - | 0.418 |
| ESCounts | **4.429** | **0.374** | **0.185** | **0.521** | **3.536** | **0.317** | **0.219** | **0.571** |

**UCFRep**. Compared to methods with frozen encoders in Tab. 1c, ESCounts with ViT-B improves the previous SoTA by +0.19 OBO and −0.17 MAE and outperforms [27] on the same RX3D101 backbone by +0.22 OBO. Our method does not outperform [63,65] that fine-tune their encoders on UCFRep. As noted in [27] this is advantageous given the dataset's size. We show this experimentally by reporting Context (F) trained from the available code of [63] with a frozen encoder, resulting in a significant performance drop with +0.51 MAE and -0.42 OBO. In all directly comparable results, ESCounts achieves stronger results.

**Qualitative Results**. In Fig. 5 we visualise predicted to ground truth counts as scatter plots. For RepCount and UCFRep, we select [18] and [63] as respective baselines and use their publicly available checkpoints[4]. ESCounts accurately predicts the number of repetitions for a wide range of counts, with most predictions being close to the ground truth i.e. the diagonal. Though predictions from both the baseline and ESCounts are close to the ground truth in low counts, they significantly diverge in high counts. We visualise specific examples and their density maps. ESCounts is robust to the magnitude of counts, with accurate predictions over low (a,b,g,k,l) and high (d,i,m) count examples. In cases of over- and under-predictions; *e.g.* (c,e,f,h,j,n) ESCounts predictions remain closer to actual counts. As shown by the density maps, ESCounts can also localise the repetitions. For Countix, even though ESCounts can predict accurate counts, as the model was trained on pseudo labels, it struggles to localise some of the repetitions. We investigate the localisation capabilities in Sec. 8 of the appendix.

[4] [27] was not used as a baseline as the code is not public. The publicly available checkpoints on Countix obtained lower results than originally reported.

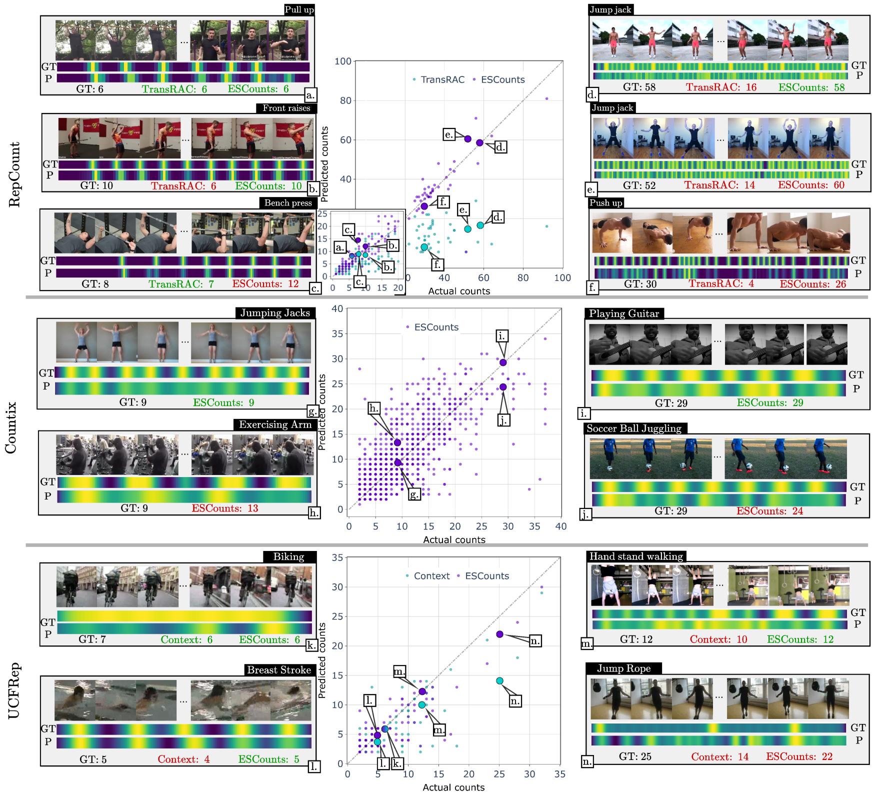


**Fig. 5: RepCount, Countix, and UCFRep scatter plot, instances, and density maps**. The dotted diagonal denotes correct predictions. We compare ESCounts against TransRAC on Repcount and Context on UCFRep. Action classes and count predictions are shown for each instance. We add the Ground Truth (GT) and Predicted (P) density maps per instance. Pseudo-labels are shown as GT for Countix.

**Cross-dataset Generalisation**. Following [18, 27], we test the generalisation capabilities of our method in Tab. 2. We use ESCounts trained on RepCount and evaluate on the Countix and UCFRep test sets. For Countix, we outperform the baseline [18] by significant margins across metrics. For UCFRep, our method surpasses [27] by $-0.21$ in MAE and $+0.22$ in OBO. ESCounts in this setting still outperforms [14, 18, 27, 28] *trained* on UCFRep in Tab. 1c, showcasing the strong ability of ESCounts to generalise to unseen actions.

### 4.3 Ablation Studies

In this section, we conduct ablation studies on RepCount [18] using ViT-B as the encoder. We study the impact of exemplars by replacing cross- with self-attention and varying the number of training exemplars. We evaluate the sensitivity of our method to the exemplar sampling probability $p$, density map $\sigma$, the impact of time-shift augmentations, and the components of our objective.

**Table 3: Ablations on RepCount** over different ESCounts settings.

**(a)** SA-only decoder

| | RMSE↓ | MAE↓ | OBZ↑ | OBO↑ |
|---|---|---|---|---|
| SA-only | 5.654 | 0.273 | 0.147 | 0.470 |
| ESCounts | **4.455** | **0.213** | **0.245** | **0.563** |

**(b)** Number of exemplars $|\mathcal{S}|$

| $\|\mathcal{S}\|$ | RMSE↓ | MAE↓ | OBZ↑ | OBO↑ |
|---|---|---|---|---|
| $\|\mathcal{S}\| = 0$ | 4.962 | 0.240 | 0.223 | 0.519 |
| $\|\mathcal{S}\| \sim \{0,1\}$ | 4.633 | 0.228 | 0.236 | 0.546 |
| $\|\mathcal{S}\| \sim \{0,2\}$ | 4.601 | 0.226 | 0.239 | 0.550 |
| $\|\mathcal{S}\| \sim \{0,1,2\}$ | **4.455** | **0.213** | 0.245 | **0.563** |
| $\|\mathcal{S}\| \sim \{0,1,2,3\}$ | 4.497 | 0.215 | **0.246** | 0.560 |
| $\|\mathcal{S}\| \sim \{0,1,2,3,4\}$ | 4.482 | 0.215 | 0.240 | 0.559 |

**(c)** Exemplar sampling

| Diff video | same class | RMSE↓ | MAE↓ | OBZ↑ | OBO↑ |
|---|---|---|---|---|---|
| ✗ | - | 4.701 | 0.224 | 0.226 | 0.521 |
| ✔ | ✗ | 5.553 | 0.270 | 0.165 | 0.464 |
| ✔ | ✔ | **4.455** | **0.213** | **0.245** | **0.563** |

**(d)** Sampling prob. $p$

| $p$ | RMSE↓ | MAE↓ | OBZ↑ | OBO↑ |
|---|---|---|---|---|
| 0.0 | 4.919 | 0.240 | 0.205 | 0.545 |
| 0.2 | 4.654 | 0.221 | 0.236 | 0.550 |
| 0.4 | **4.455** | **0.213** | **0.245** | **0.563** |
| 0.6 | 4.561 | 0.218 | 0.240 | 0.558 |
| 0.8 | 4.735 | 0.230 | 0.223 | 0.553 |
| 1.0 | 5.012 | 0.245 | 0.218 | 0.532 |

**(e)** Density peaks $\sigma$

| $\sigma$ | RMSE↓ | MAE↓ | OBZ↑ | OBO↑ |
|---|---|---|---|---|
| Variable | 6.152 | 0.301 | 0.165 | 0.457 |
| 0 | 5.145 | 0.241 | 0.206 | 0.510 |
| 0.25 | 4.871 | 0.226 | 0.228 | 0.542 |
| 0.50 | **4.455** | **0.213** | **0.245** | **0.563** |
| 0.75 | 4.683 | 0.218 | 0.240 | 0.556 |
| 1.00 | 4.732 | 0.223 | 0.238 | 0.552 |

**(f)** Timeshift Aug. $|\mathbf{K}|$

| $\|\mathbf{K}\|$ | RMSE↓ | MAE↓ | OBZ↑ | OBO↑ |
|---|---|---|---|---|
| 1 | 4.592 | 0.221 | 0.235 | 0.552 |
| 2 | 4.493 | 0.217 | 0.242 | 0.556 |
| 3 | 4.471 | 0.213 | 0.243 | 0.561 |
| 4 | **4.455** | **0.213** | **0.245** | **0.563** |

**(g)** Effect of Objective

| Obj | RMSE↓ | MAE↓ | OBZ↑ | OBO↑ |
|---|---|---|---|---|
| MSE | 5.109 | 0.273 | 0.215 | 0.532 |
| +MAE | **4.455** | **0.213** | **0.245** | **0.563** |

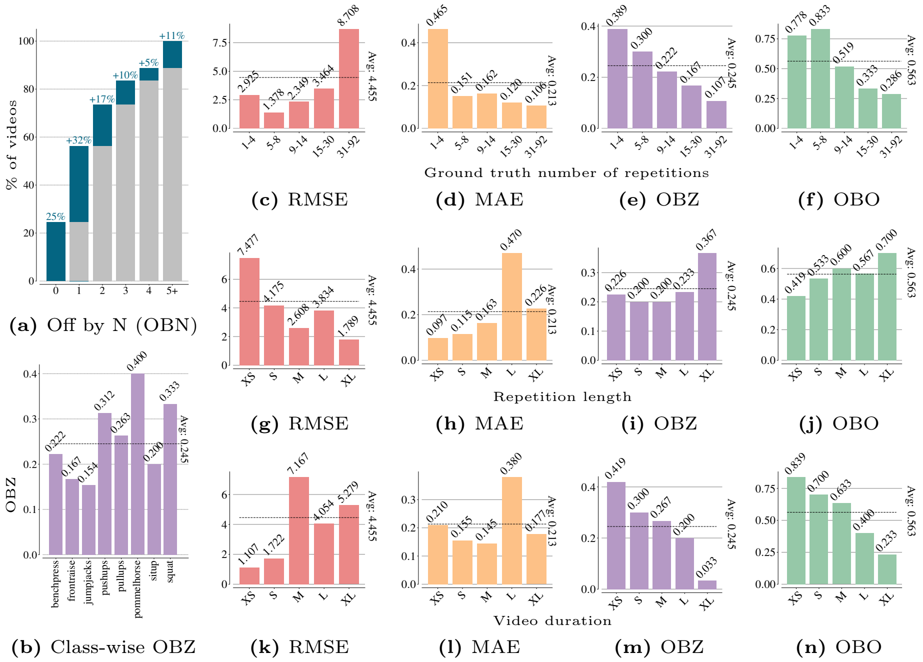
**Do exemplars help in training?** We study the impact of using exemplars for training by directly replacing the cross-attention decoder blocks with self-attention. As seen in Tab. 3a, using self-attention (SA-only) performs significantly worse than our proposed ESCounts. Cross-attending exemplars decrease the RMSE/MAE by $-1.20$ and $-0.06$ whilst improving OBZ and OBO by $+0.10$ and $+0.09$, respectively. This emphasises the benefits of exemplar-based VRC.

**How many exemplars to sample?** A varying number of training exemplars $|\mathcal{S}|$ is used in Tab. 3b. For $|\mathcal{S}| = 0$, we train **only** the zero-shot latent $\mathbf{z}_0$ alongside the model's parameters. Training with $|\mathcal{S}| \sim \{0, 1, 2\}$ provides the best zero-shot scores at inference with our method efficiently learning to generalise by attending to only a few exemplars. The inclusion of more exemplars saturates performance.

**How to sample exemplars?** In Tab. 3c we analyse the impact of sampling exemplars from the same or other training videos. As expected, keeping the same action category for both exemplar and query videos performs the best, as ensuring the same action semantics between exemplars and query video helps to learn correspondence. In this table, we used sampling probability $p = 0.4$. In Tab. 3d, we vary the sampling probability from other videos of the same underlying action $p$. For $p = 0.0$, exemplars are sampled exclusively from the query video, whilst for $p = 1.0$, exemplars are sampled solely from other videos of the same class. The best performance was observed with $p = 0.4$, showcasing that the visual characteristics of exemplars from the same video are critical for VRC compared to class semantics.

**What's the impact of time-shift augmentations?** Predictions are aggregated over $|\mathbf{K}|$ density maps by time-shifting the video input. As shown in Tab. 3f, having $|\mathbf{K}| = 4$ shifted start/end positions provides the best results. However, results are strong even without test-time augmentations in $|\mathbf{K}| = 1$.

**What should be the density map variance?** Density maps are constructed as vectors with normal distributions $\mathcal{N}(\cdot; \mu, \sigma)$ over repetition starts/ends timestamps. Reducing $\sigma$ increases the sharpness, resulting in a single delta function for $\sigma = 0$. We ablate over different $\sigma$ in Tab. 3e. Denser and successive repetitions can benefit from sharper peaks of small $\sigma$ and sparser repetitions of larger dura-



**(a)** Off by N (OBN) **(b)** Class-wise OBZ **(c)** RMSE **(d)** MAE **(e)** OBZ **(f)** OBO **(g)** RMSE **(h)** MAE **(i)** OBZ **(j)** OBO **(k)** RMSE **(l)** MAE **(m)** OBZ **(n)** OBO

**Fig. 6: Grouped VRC scores** over different number of repetitions and lengths. (a) overviews the Off by N accuracy for increasing Ns. (b) shows OBZ by action class. The first row (c–f) reports results over different counts. (g–j) reports scores over groups by repetition durations. (k–n) reports metrics grouped by video duration.

tions can benefit from large $\sigma$. We also ablate using variable $\sigma$ that changes with the duration of repetition segments. Having $\sigma = 0.5$ provides the best results with a balance between sharpness and covering the duration of repetitions.

**How helpful is the MAE for the objective?** We analyse ESCounts' performance with and without the MAE loss from [65] in Tab. 3g. The combined objective helps performance for diverse counts across all metrics.

**How close are predictions to the ground truth?** We further relax the off-by metrics to Off-By-N in Fig. 6a to visualise the proximity of predictions to the ground truth. Overall, 84% of predictions are within ±3 of the actual count.

**What is the performance per action category?** In Fig. 6b, we plot the OBZ per action class. ESCounts performs fairly uniformly across all classes with the best-performing categories being *pommelhourse* and *squat*.

**How does performance differ across counts, repetition lengths, and video durations?** Up to this point, we have focused on the performance across all videos regardless of individual attributes. We now consider the sensitivity of ESCounts across equally sized groups based on the number of repetitions, average repetition length, and video duration.

We report all metrics over groups of counts in Figs. 6c to 6f. As expected, our method performs best in groups of smaller counts with higher counts being more challenging to predict precisely.

In Figs. 6g to 6j we report VRC metrics with results grouped by average video repetition duration. These are grouped, into equal sized bins, to XS=(0-0.96)s, S=(0.96-1.53)s, M=(1.53-2.29)s, L=(2.29-3.09)s, XL=(>3.09)s. Predicting density maps is more challenging for short repetitions. However, ESCounts can still correctly predict counts across repetition lengths as shown by Figs. 6i and 6j. We also group videos by duration into XS=(8.0-11.0)s, S=(11.0, 26.0)s, M=(26.0, 33.9)s, L=(33.9-45.9)s and XL=(45.9-68.0)s. From Figs. 6k to 6n, counting repetitions from longer videos is more challenging.

### 4.4 Multi-Shot Inference

We use learnt latents for exemplar-free inference.

**Table 4: Number of shots at inference**. We test using exemplars from the same video or a different video of the same action class from the train set.

| Shots | Same video | RepCount | | | | UCFRep | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | RMSE↓ | MAE↓ | OBZ↑ | OBO↑ | RMSE↓ | MAE↓ | OBZ↑ | OBO↑ |
| 0 | N/A | 4.455 | 0.213 | 0.245 | 0.563 | 1.972 | 0.216 | 0.381 | 0.704 |
| 1 | ✗ | 4.432 | 0.207 | 0.251 | 0.563 | 1.912 | 0.211 | 0.388 | 0.712 |
| | ✔ | 4.369 | 0.210 | 0.247 | 0.589 | 1.890 | 0.203 | 0.400 | 0.714 |
| 2 | ✗ | 4.384 | **0.206** | 0.251 | 0.572 | 1.885 | 0.208 | 0.391 | 0.720 |
| | ✔ | 4.360 | 0.209 | 0.247 | 0.592 | 1.857 | 0.199 | 0.419 | 0.718 |
| 3 | ✗ | 4.381 | 0.207 | **0.252** | 0.579 | 1.878 | 0.207 | 0.399 | **0.730** |
| | ✔ | **4.351** | **0.206** | 0.250 | **0.596** | **1.855** | **0.198** | **0.420** | 0.723 |

Prior object counting [32, 37] report results with exemplars (i.e. object crops) at inference. While this is not comparable to other VRC works, we can assess our method's ability to utilise exemplars during inference in Tab. 4. Video exemplars steadily improve performance as the number of exemplars increases. Our model cross-attends exemplars in parallel, training with 0–2 exemplars, and can even use $>2$ exemplars at inference. We show comparable results when sampling exemplars from the test video or training videos with the same action category. Combined with a classifier, a closed-set approach can be envisaged that classifies the action and then sources exemplars from the training set to assist counting during inference.

## 5 Conclusion

We have proposed to utilise exemplars for video repetition counting. We introduce Every Shot Counts (ESCounts), an attention-based encoder-decoder that learns to correspond exemplar repetitions across a full video. We define a learnable zero-shot latent that learns representations of generic repetitions, to use during inference. Extensive evaluation on RepCount, Countix, and UCFRep demonstrates the merits of ESCounts achieving state-of-the-art results on the traditional MAE and OBO metrics and the newly introduced RMSE and OBZ. We provide detailed analysis and ablations of our method, highlighting the importance of training with exemplars and time-shift augmentations. The diversity of these exemplars is an aspect for future exploration.

**Acknowledgements.** This work uses publicly available datasets and annotations for results and ablations. Research is supported by EPSRC UMPIRE (EP/T004991/1). S. Sinha is supported by EPSRC DTP studentship.

## References

1. Alayrac, J.B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al.: Flamingo: a Visual Language Model for Few-Shot Learning. In: Advances in Neural Information Processing Systems (NeurIPS). pp. 23716–23736 (2022)
2. Albu, A.B., Bergevin, R., Quirion, S.: Generic Temporal Segmentation of Cyclic Human Motion. Pattern Recognition **41**(1), 6–21 (2008)
3. Amini-Naieni, N., Amini-Naieni, K., Han, T., Zisserman, A.: Open-world Text-specified Object Counting. In: British Machine Vision Conference (BMVC). p. 510 (2023)
4. Arteta, C., Lempitsky, V., Zisserman, A.: Counting in the Wild. In: European Conference on Computer Vision (ECCV). pp. 483–498 (2016)
5. Azy, O., Ahuja, N.: Segmentation of Periodically Moving Objects. In: International Conference on Pattern Recognition (ICPR). pp. 1–4 (2008)
6. Bacharidis, K., Argyros, A.: Repetition-aware Image Sequence Sampling for Recognizing Repetitive Human Actions. In: International Conference on Computer Vision Workshops (ICCVw). pp. 1878–1887 (2023)
7. Briassouli, A., Ahuja, N.: Extraction and Analysis of Multiple Periodic Motions in Video Sequences. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) **29**(7), 1244–1261 (2007)
8. Carreira, J., Zisserman, A.: Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6299–6308 (2017)
9. Chao, Y.W., Vijayanarasimhan, S., Seybold, B., Ross, D.A., Deng, J., Sukthankar, R.: Rethinking the Faster R-CNN Architecture for Temporal Action Localization. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1130–1139 (2018)
10. Chattopadhyay, P., Vedantam, R., Selvaraju, R.R., Batra, D., Parikh, D.: Counting Everyday Objects in Everyday Scenes. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1135–1144 (2017)
11. Cholakkal, H., Sun, G., Khan, F.S., Shao, L.: Object Counting and Instance Segmentation with Image-level Supervision. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 12397–12405 (2019)
12. Cutler, R., Davis, L.S.: Robust Real-Time Periodic Motion Detection, Analysis, and Applications. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) **22**(8), 781–796 (2000)
13. Destro, M., Gygli, M.: CycleCL: Self-supervised Learning for Periodic Videos. In: Winter Conference on Applications of Computer Vision (WACV). pp. 2861–2870 (2024)
14. Dwibedi, D., Aytar, Y., Tompson, J., Sermanet, P., Zisserman, A.: Counting Out Time: Class Agnostic Video Repetition Counting in the Wild. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10387–10396 (2020)
15. Feichtenhofer, C., Li, Y., He, K., et al.: Masked Autoencoders as Spatiotemporal Learners. In: Advances in Neural Information Processing Systems (NeurIPS). pp. 35946–35958 (2022)
16. Gaidon, A., Harchaoui, Z., Schmid, C.: Temporal Localization of Actions with Actoms. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) **35**(11), 2782–2795 (2013)

17. Hsieh, M.R., Lin, Y.L., Hsu, W.H.: Drone-based Object Counting by Spatially Regularized Regional Proposal Network. In: International Conference on Computer Vision (ICCV). pp. 4145–4153 (2017)
18. Hu, H., Dong, S., Zhao, Y., Lian, D., Li, Z., Gao, S.: TransRAC: Encoding Multi-scale Temporal Correlation with Transformers for Repetitive Action Counting. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 19013–19022 (2022)
19. Huang, J., Li, Y., Feng, J., Wu, X., Sun, X., Ji, R.: Clover: Towards A Unified Video-Language Alignment and Fusion Model. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 14856–14866 (2023)
20. Jain, M., Ghodrati, A., Snoek, C.G.: ActionBytes: Learning From Trimmed Videos to Localize Actions. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1171–1180 (2020)
21. Jiang, R., Liu, L., Chen, C.: CLIP-Count: Towards Text-Guided Zero-Shot Object Counting. In: ACM International Conference on Multimedia (MM). pp. 4535–4545 (2023)
22. Laptev, I., Belongie, S.J., Pérez, P., Wills, J.: Periodic Motion Detection and Segmentation via Approximate Sequence Alignment. In: International Conference on Computer Vision (ICCV). pp. 816–823 (2005)
23. Laradji, I.H., Rostamzadeh, N., Pinheiro, P.O., Vazquez, D., Schmidt, M.: Where are the Blobs: Counting by Localization with Point Supervision. In: European Conference on Computer Vision (ECCV). pp. 547–562 (2018)
24. Lempitsky, V., Zisserman, A.: Learning to Count Objects in Images. In: Advances in Neural Information Processing Systems (NeurIPS) (2010)
25. Levy, O., Wolf, L.: Live Repetition Counting. In: International Conference on Computer Vision (ICCV). pp. 3020–3028 (2015)
26. Li, M., Zhang, Z., Huang, K., Tan, T.: Estimating the Number of People in Crowded Scenes by MID Based Foreground Segmentation and Head-shoulder Detection. In: International Conference on Pattern Recognition (ICPR). pp. 1–4 (2008)
27. Li, X., Xu, H.: Repetitive Action Counting With Motion Feature Learning. In: Winter Conference on Applications of Computer Vision (WACV). pp. 6499–6508 (2024)
28. Li, Z., Ma, X., Shang, Q., Zhu, W., Ci, H., Qiao, Y., Wang, Y.: Efficient action counting with dynamic queries. arXiv preprint arXiv:2403.01543 (2024)
29. Lin, H., Hong, X., Wang, Y.: Object Counting: You Only Need to Look at One. arXiv preprint arXiv:2112.05993 (2021)
30. Lin, K.Q., Zhang, P., Chen, J., Pramanick, S., Gao, D., Wang, A.J., Yan, R., Shou, M.Z.: UniVTG: Towards Unified Video-Language Temporal Grounding. In: International Conference on Computer Vision (ICCV). pp. 2794–2804 (2023)
31. Lin, T., Zhao, X., Su, H., Wang, C., Yang, M.: BSN: Boundary Sensitive Network for Temporal Action Proposal Generation. In: European conference on computer vision (ECCV). pp. 3–19 (2018)
32. Liu, C., Zhong, Y., Zisserman, A., Xie, W.: Countr: Transformer-based generalised visual counting. In: British Machine Vision Conference (BMVC). p. 370 (2022)
33. Liu, C., Weng, X., Mu, Y.: Recurrent Attentive Zooming for Joint Crowd Counting and Precise Localization. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1217–1226 (2019)
34. Liu, Z., Ning, J., Cao, Y., Wei, Y., Zhang, Z., Lin, S., Hu, H.: Video Swin Transformer. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3202–3211 (2022)

35. Long, F., Yao, T., Qiu, Z., Tian, X., Luo, J., Mei, T.: Gaussian Temporal Awareness Networks for Action Localization. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 344–353 (2019)
36. Lu, C., Ferrier, N.J.: Repetitive Motion Analysis: Segmentation and Event Classification. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) **26**(2), 258–263 (2004)
37. Lu, E., Xie, W., Zisserman, A.: Class-agnostic Counting. In: Asian Conference on Computer Vision (ACCV). pp. 669–684 (2019)
38. Mangalam, K., Akshulakov, R., Malik, J.: EgoSchema: A Diagnostic Benchmark for Very Long-form Video Language Understanding. In: Advances in Neural Information Processing Systems (NeurIPS) (2023)
39. Noroozi, M., Pirsiavash, H., Favaro, P.: Representation Learning by Learning to Count. In: International Conference on Computer Vision (ICCV). pp. 5898–5906 (2017)
40. Oñoro-Rubio, D., López-Sastre, R.J.: Towards perspective-free object counting with deep learning. In: European Conference on Computer Vision (ECCV). pp. 615–629 (2016)
41. Panagiotakis, C., Karvounas, G., Argyros, A.: Unsupervised Detection of Periodic Segments in Videos. In: International Conference on Image Processing (ICIP). pp. 923–927 (2018)
42. Pogalin, E., Smeulders, A.W., Thean, A.H.: Visual Quasi-Periodicity. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1–8 (2008)
43. Pramanick, S., Song, Y., Nag, S., Lin, K.Q., Shah, H., Shou, M.Z., Chellappa, R., Zhang, P.: EgoVLPv2: Egocentric Video-Language Pre-training with Fusion in the Backbone. In: International Conference on Computer Vision (ICCV). pp. 5285–5297 (2023)
44. Ramakrishnan, S.K., Al-Halah, Z., Grauman, K.: NaQ: Leveraging Narrations as Queries to Supervise Episodic Memory Supplementary Materials. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6694–6703 (2023)
45. Runia, T.F., Snoek, C.G., Smeulders, A.W.: Real-World Repetition Estimation by Div, Grad and Curl. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9009–9017 (2018)
46. Seguí, S., Pujol, O., Vitria, J.: Learning to Count with Deep Object Features. In: Conference on Computer Vision and Pattern Recognition Workshops (CVPRw). pp. 90–96 (2015)
47. Shi, M., Lu, H., Feng, C., Liu, C., Cao, Z.: Represent, Compare, and Learn: A Similarity-Aware Framework for Class-Agnostic Counting. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9529–9538 (2022)
48. Shou, Z., Wang, D., Chang, S.F.: Temporal Action Localization in Untrimmed Videos via Multi-stage CNNs. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1049–1058 (2016)
49. Slaughter, V., Itakura, S., Kutsuki, A., Siegal, M.: Learning to Count Begins in Infancy: Evidence from 18 Month Olds' Visual Preferences. Proceedings of the Royal Society B: Biological Sciences **278**(1720), 2979–2984 (2011)
50. Soomro, K., Zamir, A.R., Shah, M.: UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild. arXiv preprint arXiv:1212.0402 (2012)
51. Thangali, A., Sclaroff, S.: Periodic motion detection and estimation via space-time sampling. In: Workshop on Applications of Computer Vision (WACV). pp. 176–182 (2005)
52. Wang, J., Feigenson, L.: Infants Recognize Counting as Numerically Relevant. Developmental science **22**(6), e12805 (2019)

53. Wang, J., Feigenson, L.: What Aspects of Counting Help Infants Attend to Numerosity? Infancy **28**(2), 218–239 (2023)
54. Wang, L., Yung, N.H.: Crowd Counting and Segmentation in Visual Surveillance. In: International Conference on Image Processing (ICIP). pp. 2573–2576 (2009)
55. Wang, Y., Li, K., Li, Y., He, Y., Huang, B., Zhao, Z., Zhang, H., Xu, J., Liu, Y., Wang, Z., et al.: InternVideo: General Video Foundation Models via Generative and Discriminative Learning. arXiv preprint arXiv:2212.03191 (2022)
56. Weinland, D., Boyer, E.: Action Recognition using Exemplar-based Embedding. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1–7 (2008)
57. Willems, G., Becker, J.H., Tuytelaars, T., Van Gool, L.: Exemplar-based Action Recognition in Video. In: British Machine Vision Conference (BMVC). pp. 3–7 (2009)
58. Wu, G., Lin, J., Silva, C.T.: IntentVizor: Towards Generic Query Guided Interactive Video Summarization. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10503–10512 (2022)
59. Xiong, H., Lu, H., Liu, C., Liu, L., Cao, Z., Shen, C.: From Open Set to Closed Set: Counting Objects by Spatial Divide-and-Conquer. In: International Conference on Computer Vision (ICCV). pp. 8362–8371 (2019)
60. Xu, J., Le, H., Nguyen, V., Ranjan, V., Samaras, D.: Zero-Shot Object Counting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 15548–15557 (2023)
61. Yao, Z., Cheng, X., Zou, Y.: PoseRAC: Pose Saliency Transformer for Repetitive Action Counting. arXiv preprint arXiv:2303.08450 (2023)
62. Ye, Q., Xu, G., Yan, M., Xu, H., Qian, Q., Zhang, J., Huang, F.: HiTeA: Hierarchical Temporal-Aware Video-Language Pre-training Supplementary Material. In: International Conference on Computer Vision (ICCV). pp. 15405–15416 (2023)
63. Zhang, H., Xu, X., Han, G., He, S.: Context-Aware and Scale-Insensitive Temporal Repetition Counting. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 670–678 (2020)
64. Zhang, Y., Zhou, D., Chen, S., Gao, S., Ma, Y.: Single-Image Crowd Counting via Multi-Column Convolutional Neural Network. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 589–597 (2016)
65. Zhang, Y., Shao, L., Snoek, C.G.: Repetitive Activity Counting by Sight and Sound. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 14070–14079 (2021)
66. Zhao, Z., Huang, X., Zhou, H., Yao, K., Ding, E., Wang, J., Wang, X., Liu, W., Feng, B.: Skim then focus: Integrating contextual and fine-grained views for repetitive action counting. arXiv preprint arXiv:2406.08814 (2024)