Tracking by Predicting 3-D Gaussians Over Time

Anonymous Author(s)
Affiliation
Address
email

Abstract

In this work, we introduce Video Gaussian Masked Autoencoders (*Video-GMAE*), a self-supervised video pretraining approach with temporal correspondences. Traditional video pretraining methods operate on patch-level tokens, limiting the model's ability to learn explicit correspondences. Our method employs Gaussian representations as intermediate tokens, enabling the explicit modeling of correspondences through self-supervision. By predicting Gaussians over time within a masked autoencoder framework, we enforce temporal consistency across video frames. With this correspondence-aware pretraining, our pretrained models were able to do *any point tracking* in zero-shot. With small-scale finetuning, our models achieve 4.16% improvement on Kinetics, 37.4% on DAVIS, and 13.1% on Kubric datasets, surpassing existing state-of-the-art self-supervised video models. To foster further research, we will release our models and code.

1 Introduction

Each pixel's journey through a video tells a story of motion. By tracking many such pixels, we can understand the structure of the scene and interaction among its constituents. Smooth-pursuit tracking emerges in children as a fundamental visual skill in the first 2-4 months of their lives, prior to acquiring 3-D understanding, sensorimotor control and object permanence [1]. Similarly, in computer vision systems, solving correspondence is essential for computational photography, 3-D understanding, and other long-range reasoning tasks. While pixel tracking has been studied extensively in vision literature [2–5], it has been limited to training models on annotated point tracks, bounding boxes, and segmentation masks[4, 6, 5], synthetic datasets [7] or specialized architectures [8–10]. In this paper, we present a self-supervised approach on videos, which learns strong representation for correspondence.

We find that representations of existing video self-supervised learning (SSL) approaches do not perform well on point tracking. We hypothesize that the classic (space-)time patch prediction objective does not strongly enforce temporal consistency, that is, this objective can be optimized without understanding pixel-level correspondence across a long sequence of frames.

How can we set up our SSL task such that this correspondence emerges in the representations? *We note that motion of objects in 3-D manifests as point tracking on the image plane. In this paper, we leverage this insight to learn point tracking by pre-training on unlabeled videos*. We train an Masked Autoencoder [11]- style encoder-decoder architecture that takes video as input and predicts *Gaussian primitives* [12] that move in time. We call our approach *Video-GMAE*, in which for the first frame of the video, we predict a fixed number of Gaussian primitives. For subsequent frames, we predict the *translation* and *color* change of each Gaussian with respect to the previous frame. In this way, each Gaussian preserves its identity over the span of N frames. Encoding a frame sequence as a set of moving Gaussians explicitly imposes temporal correspondence in 3-D as an inductive bias in our training. This inductive bias makes the SSL task *harder*, inducing the latent representations to encode long-term correspondence.

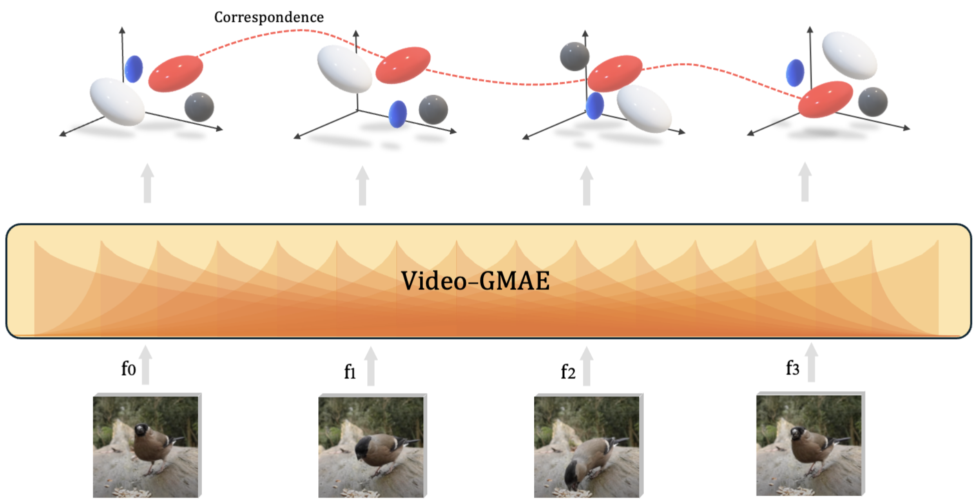

Figure 1: **Self-supervised Video Pretraining for Correspondence:** Given a sequence of video frames, our approach *Video-GMAE* predicts Gaussian primitives, for each frame to reconstruct the whole video. In addition to this, we also enforce correspondence in the Gaussian primitives by predicting the changes in the Gaussian primitives for the frames except the first frame.

To investigate our pretraining methodology, we devise an algorithm to compute point tracks from the predicted 4-D Gaussians *zero-shot*. We find that a zero-shot tracker naturally emerges from our correspondence-aware Gaussian representations. *Video-GMAE* also gives latent representations that are useful for tracking. Finetuning *Video-GMAE* on tracking datasets outperforms prior supervised and self-supervised methods. *Video-GMAE* also outperforms other video SSL approaches such as VideoMAE [13], MAE-ST [14] on frozen encoder tracking evaluation.

Broadly, we aim to bring classic end-to-end SSL together with differentiable rendering. Differentiable rendering is a natural inductive bias for video SSL: since videos are basically 2-D projections of a 3-D world, dynamic yet consistent over time. We hope that this work aids the search for video-SSL methods that learn more long-horizon and more generalizable representations.

2 Related Works

Self-supervised Learning: Over the years, self-supervised pretraining has shown strong performance across domains like language, vision, and robotics. In computer vision, there are broadly two schools of thought: discriminative and reconstructive pretraining. Discriminative methods typically train a model to recognize that different augmentations of the same image should be close in feature space. Early works like [15] and SimCLR [16] demonstrated that contrastive learning over such instance discrimination tasks can yield strong visual features. Later methods like MoCo [17] and DINO [18] further pushed this line of work, showing that such learned representations can transfer well to a range of downstream tasks.

Reconstructive pre-training learns to model the data distribution by trying to reconstruct an image or a video from its noisy version. One of the most successful methods for such pre-training in computer vision has been the BERT [19]-style masked modeling of images proposed by BEiT [20], and MAE [11]. Compared to BERT, MAE uses asymmetric encoder-decoders, allowing it to be very efficient at training with high masking ratios. This style of reconstructive pre-training learns strong visual priors and shows impressive results on various downstream tasks such as object detection [21], pose estimation [22], and robot tasks [23]. Extending this to videos, VideoMAE [24] and MAE-ST [14] showed that, with large masking ratios, masked auto-encoding can learn very strong representations from unlabeled videos. Another line of generative video modeling uses autoregressive models to simply predict the next patch or next frame [25, 26]. Recently, these models are used as an encoder for vision-language models [27, 28].

Gaussian Splatting: Gaussian Splatting [12] is a recent differentiable rendering method that uses 3-D Gaussian primitives as the underlying representation, allowing for flexible optimization and high-quality reconstructions. This idea builds on a broader trend in differentiable rendering, which has become a popular way to connect 3-D geometry with 2-D image supervision. By making the rendering process differentiable, these methods enable gradient-based learning of 3-D structures, like meshes and point clouds, from images. For instance, [29] introduced a soft rasterizer for mesh-based rendering, while [30] proposed an efficient differentiable renderer for large point clouds. NeRF [31] and Mip-NeRF [32] extend this idea to volumetric scene representations, using differentiable volume rendering [33] to learn 3-D radiance fields from just a few multi-view images.

Point Tracking: Tracking has been studied in computer vision under different scales. Traditional methods like the Kanade–Lucas–Tomasi tracker [34] have been widely used for this purpose, leveraging local image gradients to track features over time. Recent deep-learning driven advancements have introduced more robust and flexible approaches. For instance, RAFT [7] extracts per-pixel features per-frame and uses correlation across frames to compute point tracks. The Tracking Any Point (TAP) [2] paradigm focuses on tracking arbitrary points on deformable surfaces, accommodating complex motions and occlusions. Models like TAPIR [35] enhance this capability by employing per-frame initialization and temporal refinement strategies, enabling accurate tracking of points across diverse scenarios. Both the above methods are trained using supervised learning with synthetic data. Another line of work focuses on developing self-supervised learning (SSL) algorithms for point tracking. CRW [36] models the evolution of patches over time as a random walk parameterized by a learnable matrix, trained using cycle consistency. GMRW [37] further extends it to pixel-level tracking. DIFT [38] shows that nearest neighbour on patch-level features extracted from pretrained diffusion models provides not just temporal, but also semantic, correspondence across frames.

Benchmark datasets like TAP-Vid [2] and TAPVid-3-D [39] have been developed to evaluate and compare the performance of various point tracking methods, providing standardized metrics and diverse testing scenarios. Overall, the evolution of point tracking techniques—from classical methods to modern deep learning-based approaches—reflects the ongoing efforts to achieve more accurate, robust, and versatile tracking capabilities in computer vision applications.

3 Preliminaries

3.1 Self-supervised Masked Autoencoders

Masked autoencoders learn data representations by randomly masking parts of the input and training the model to predict the missing content. In language, BERT [40] follows this approach by masking some text tokens and using a transformer [41] to predict them. For images, methods like MAE [42] and BEiT [43] mask image patches and train the model to reconstruct the missing regions. In videos, approaches such as VideoMAE [13] and MAE-ST [44] extend this idea by masking spatiotemporal patches across frames. Specifically, MAE-ST uses a Vision Transformer (ViT) [45] encoder to process the visible patches, while a lightweight ViT decoder takes both visible and masked tokens to reconstruct the missing video content.

3.2 3D Gaussian Splatting

3-D Gaussian Splatting originally introduced for optimization-based single-scene 3-D reconstruction [46], and later extended to image-level representation learning in [47]. Each primitive is defined by a 3D center position $p \in \mathbb{R}^3$, a covariance matrix $\Sigma \in \mathbb{R}^{3\times3}$, a color $r \in \mathbb{R}^3$, and an opacity value $o \in \mathbb{R}$, collectively encoding the geometry and appearance of the scene. During rendering, the Gaussians are transformed into the camera frame and projected onto the image plane using volumetric splatting. This rendering pipeline is fully differentiable, allowing gradients to flow back to the Gaussian parameters from the rendered output. Following standard practice, the covariance matrix is factorized as $\Sigma = RSS^TR^T$, where $S = \mathrm{diag}(s) \in \mathbb{R}^{3\times3}$ is a diagonal scaling matrix parameterized by $s \in \mathbb{R}^3$, and $R \in \mathbb{R}^{3\times3}$ is a rotation matrix represented via a quaternion $\phi \in \mathbb{R}^4$. As a result, each Gaussian is fully described by a 14-dimensional vector $g = \{p, s, \phi, r, o\} \in \mathbb{R}^{14}$.

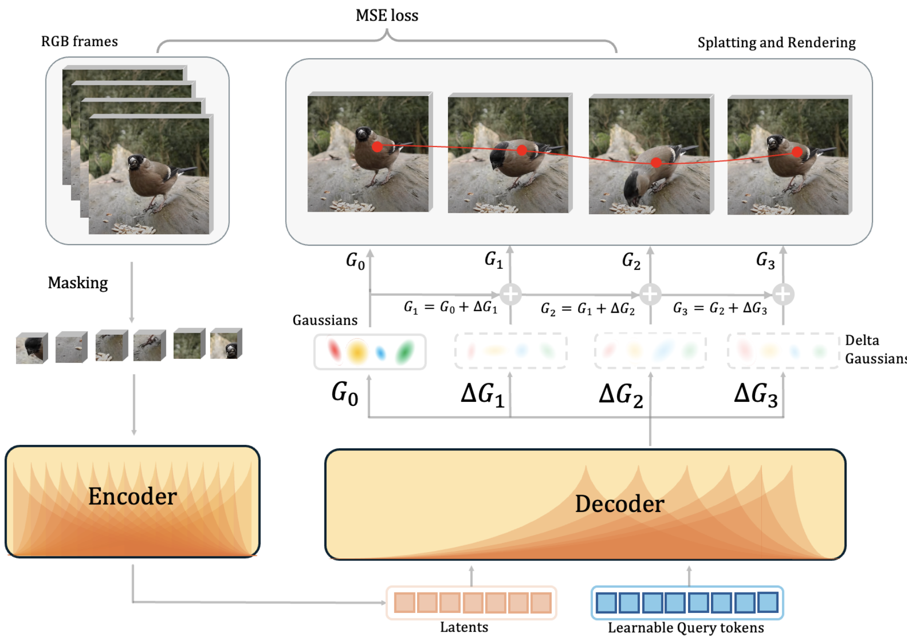

Figure 2: **Video Masked Auto encoding via Gaussian Splatting:** The ViT Encoder processes masked input frames to produce latent embeddings. The ViT Decoder then predicts explicit Gaussian parameters for frame f_1 based on query tokens, including color, opacity, center, scale, and orientation, and the Gaussian deltas for the rest of the frames. The explicit Gaussians for frame f_1 to f_t are calculated and rendered via differentiable volume splitting to reconstruct all the frames. We pre-train our models fully end-to-end with self-supervision.

4 Self-Supervised Pretraining

Our model is trained as a video masked autoencoder on the Kinetics dataset [48]. First, we mask 95% of the video patches, from k frames, and the encoder sees only the masked patches and generates intermediate latents. Then we concatenate the latents with the decoder query tokens. From this, the decoder generates $k \times n$ Gaussians for rendering. Only the first n Gaussians are full Gaussians, and the rest are delta Gaussians, which are added to the first previous frame Gaussians iteratively. See Fig 4 how we create Gaussians for the rest of the frames, by adding the delta Gaussians to the previous Gaussians. By doing this, residual prediction we enforce correspondence in Gaussians over time.

To pretrain on videos, we start with an image-based encoder trained to predict 3-D Gaussians for an image [47]. Then we extend this model to a video encoder by adding a separate learnable positional embedding for the time axis. We train this video model on k frames, with each frame having 16×16 patches. We do patchification at the frame level to utilize the image model– GMAE [47]. We train this model with high masking ratio of 95% as in [14].

On the decoder, we have $k \times n$ learnable query tokens, $k = 16$ frames at pretraining and $n = 256$ Gaussians per frame. Let's assume G_0 is the set Gaussians for the first frame $G_0 = \{g_1, g_2, ...g_n\}$. For the following frames, the decoder only predicts delta Gaussians, $\Delta G_1, \Delta G_2, ..., \Delta G_k$. To train our model, we iteratively update Gaussians for each frame and render them to get reconstructed video. This video is then used to train the model with reconstruction loss.

$$G_t = \Delta G_t + G_{t-1} \tag{1}$$

Here, G_0 and ΔG_t are predictions from the decoder. Then we render all the k sets of Gaussians for each frame and train the encoder and the decoder with reconstruction loss. This formulation allows us to train the model with correspondence under self-supervised training.

5 Zero-shot Point Tracking

In this section, we present an algorithm to extract point tracks from the collection of Gaussian primitive trajectories predicted by *Video-GMAE*. At a high level, we splat the Gaussian primitives on the image plane with the RGB values of each Gaussian replaced with the projected 2-D Gaussian mean deltas. This converts Δp_i, a vector in 3-D space, to a 2-D line segment on the image plane. Following this line for any point tracks it to the next frame.

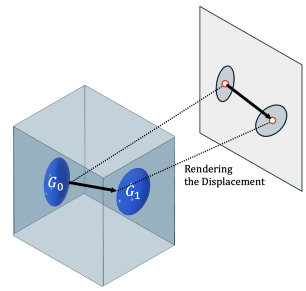

Figure 3: **Zero-shot Point Tracking:** The 3-D centers of the predicted Gaussian primitives are projected and the subsequent 2-D displacement vector is rendered.

Given an initial set of Gaussians $G_0 = \{g_1, g_2, \ldots, g_n\}$ for frame $t = 0$, each primitive $g_i = \{p_i^{(0)}, s_i, \phi_i, r_i^{(0)}, o_i\}$ contains a 3-D position $p_i \in \mathbb{R}^3$. For subsequent frames, the model predicts residual updates $\Delta G_t = \{\Delta p_i^{(t)}, \Delta r_i^{(t)}\}_{i=1}^n$ to the Gaussian means. We apply these deltas recursively to get the means of the Gaussian primitives as:

$$p_i^{(t+1)} = p_i^{(t)} + \Delta p_i^{(t)}. \tag{2}$$

To extract motion in the image plane, we project the means of each Gaussian into pixel coordinates using the same camera intrinsics $K \in \mathbb{R}^{3\times 3}$ and extrinsics $[R \mid t] \in \mathrm{SE}(3)$ we use to render during the training phase. Let $\Pi(\cdot)$ denote the perspective projection:

$$x_i^{(t)} = \Pi\left(K[R \mid t], p_i^{(t)}\right), \quad x_i^{(t+1)} = \Pi\left(K[R \mid t], p_i^{(t+1)}\right). \tag{3}$$

We calculate the instantaneous 2-D displacement vector $d_i^{(t)}$ carried by each Gaussian, encode them as pseudo-RGB values $c_i^{(t)} = (d_{i,x}^{(t)}, d_{i,y}^{(t)}, 0)$, and and splat them onto the image plane using standard volumetric alpha compositing. The resulting dense flow field $F^{(t)} \in \mathbb{R}^{H\times W\times 2}$ at each pixel $u \in \mathbb{R}^2$ is computed by an opacity-weighted average of all displacements:

$$F^{(t)}(u) = \sum_{i=1}^{n} \alpha_i^{(t)}(u), d_i^{(t)}, \tag{4}$$

where $\alpha_i^{(t)}(u) \in [0, 1]$ is the rasterized visibility of Gaussian i in the pixel u, calculated using differentiable Gaussian splatting [46]. To track a point $p^{(0)} \in \mathbb{R}^2$ forward through time, we advect it using bilinear-interpolated flow from the generated sequence.

$$p^{(t+1)} = \text{clip}\left(p^{(t)} + \text{bilinear}\left(F^{(t)}, p^{(t)}\right)\right) \tag{5}$$

where $\text{clip}(\cdot)$ ensures the point remains within image bounds. This enables robust tracking of arbitrary query points with no supervision.

6 Supervised Finetuning for Point Tracking

We evaluate the encoder's learned representation on point tracking to assess the improved correspondence-based features. For point-tracking evaluation, we adopt a cross-attention–based readout architecture, and evaluate on the TAP-Vid protocol which includes three datasets: TAP-Vid-Kinetics[2], TAP-Vid-DAVIS[2], and Kubric[49]. Each evaluation run uses a single query per point track. We follow the evaluation protocol described in [37]. We either freeze the encoder for the frozen results or fine-tune the encoder for the fine-tune results, and the readout network consumes the spatio-temporal encoder features to predict tracked points across frames.

Following the design in [50], we first apply layer normalization to the encoder features and add learned temporal embeddings. We use 64-dimensional fourier-based positional queries [50], projected to the token feature dimension. These queries cross-attend to the processed encoder features using a 16-headed attention mechanism. This is followed by a residual MLP with hidden size four times the token feature dimension and GeLU [51] activation, and finally a linear layer mapping followed

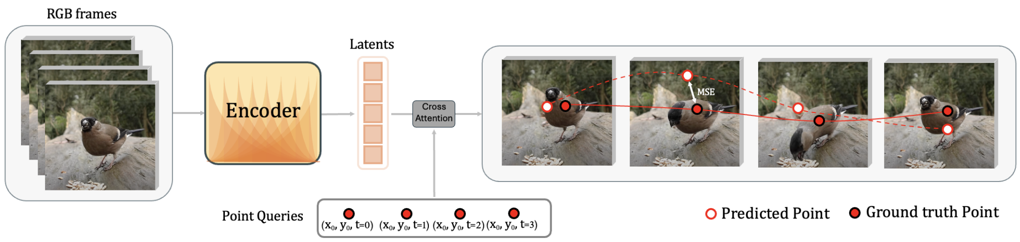

Figure 4: **Finetuning for Point Tracking:** To get the best point tracks, we use our pretrained encoder without masking, and query the latents to predict the point tracks. We finetune this model, using annotated Kubric [49] dataset. The fourier embeddings of the initial queries are calculated, and they cross-attend to the encoder latents in the fine-tuned cross-attention readout layer.

by sigmoid activation to a 3-D vector corresponding to the tracked 2-D point in each frame and its occlusion.

For frozen evaluation, we train only the readout layers. Each model is trained on one A100 with a batch size of 8 and 16 tracks per video for 50k steps using the AdamW optimizer (weight decay 5e-2) and a learning rate sweep over { 5e-5, 1e-4, 3e-4 }, with the best-performing configuration selected. In the finetuned setting, the same setup is used, except both the encoder and the readout are trained end-to-end.

7 Experiments

Evaluation: For point-tracking evaluation, we report three metrics that collectively measure accuracy, robustness, and occlusion handling. Average Jaccard Score (AJ) computes the intersection-over-union between predicted and ground-truth visible regions, reflecting alignment quality over time. Average Points within Threshold (δ^x_{avg}) reports the percentage of predictions falling within a small pixel radius of the ground truth, offering an interpretable precision metric. Occlusion Accuracy (OA) quantifies how often the model correctly predicts whether a point is visible or occluded in each frame. We conduct a series of experiments to evaluate the effectiveness and design choices of our proposed model. Unless otherwise specified, all experiments are performed with a frozen encoder, focusing exclusively on optimizing the lightweight decoder.

Implementation Details: We pretrained the models on 64 V100s for 90 epochs with a batch size of 128, a learning rate of 1e-3, using AdamW optimizer (weight decay 5e-2). We utilized 2000 warm-up steps and cosine decay for the learning rate. We also employ gradient clipping with a value of 2.0.

7.1 Zero-shot Tracking Results

We compute AJ and δ^x_{avg} for zero-shot tracking on all three datasets, similar to the fine-tuned point-tracking evaluation following [2]. Table 2 shows the zero-shot tracking metrics. Figure 5 shows some qualitative examples of the zero-shot tracking. Our zero-shot tracking results on Kubric are comparable to DIFT-D [38] numbers, our TAP-Vid Davis results are comparable with Flow-Walk-D [52] and GMRW-D [37]. Since we pretrained on Kinetics, our zero-shot tracking results outperform all self-supervised methods. The zero-shot results show the benefit of our self-supervised approach, which is comparable to other self-supervised approaches that can not scale.

7.2 Comparison with Video Pretraining Methods

We compare *Video-GMAE* with existing state-of-the-art self-supervised pretraining methods on the task of tracking. Specifically, we compare against VideoMAE [13] and MAE-ST [44]. Following the encoder and decoder configurations of the MAE Base and MAE Large architectures, we train two versions of our model: *Video-GMAE*-base and *Video-GMAE*-large, respectively. These are compared against VideoMAE and MAE-ST using the same frozen-encoder training setup. Table 1 contains our results. *Video-GMAE* outperforms the other pretraining baselines on all datasets.

Figure 5: **Zero shot Qualitative Results:** We show qualitative results (16 frames, we only show every other frame here) from our pretrained *Video-GMAE*-base model. Without any track labels, the model was able to track objects with camera motion and pose changes, and shows robust tracking over long videos.

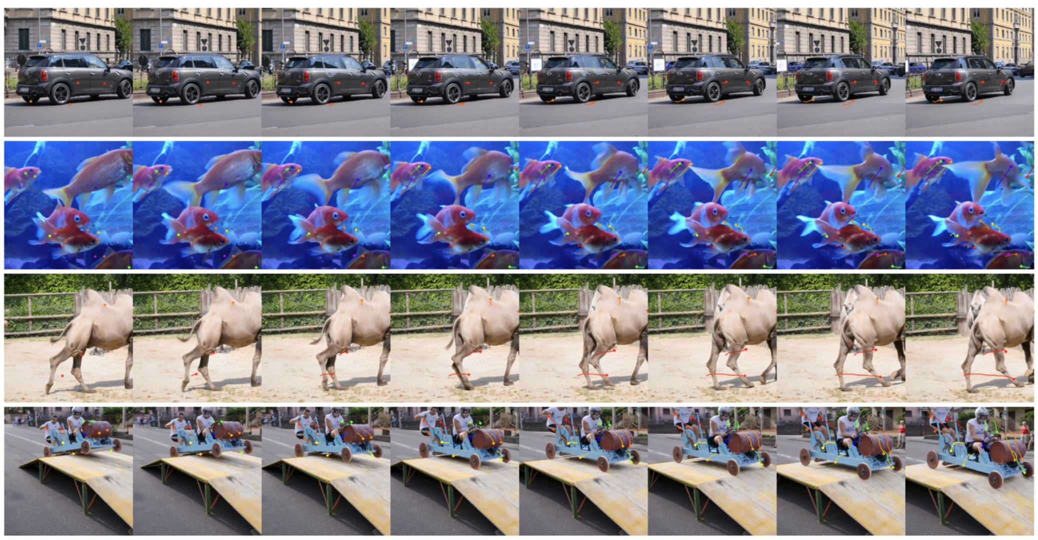

Figure 6: **Qualitative Results from Finetuned Model:** We show qualitative results (16 frames, we only show every other frame here) from our finetuned *Video-GMAE*-base model, after finetuning on Kubric datasets [49]. With small finetuning, our model was able to achieve state-of-the-art performance on point tracking, and was able to tracking points over long range with high precision.

Model	**Kinetics**			**DAVIS**			**Kubric**		
	AJ↑	δ^x_{avg} ↑	OA↑	AJ↑	δ^x_{avg} ↑	OA↑	AJ↑	δ^x_{avg} ↑	OA↑
MAE-ST[14]	17.7	24.3	90.2	9.14	14.0	81.6	18.7	26.4	88.8
VideoMAE[13]	21.1	28.8	89.5	12.0	18.7	80.7	20.0	29.5	87.6
Video-GMAE	**28.4**	**36.2**	**92.5**	**19.0**	**26.5**	**86.2**	**32.8**	**43.2**	**89.8**

Table 1: Cross-dataset comparison of pre-trained backbones (all are with a frozen encoder) comparing *Video-GMAE* to VideoMAE and MAE-ST. Even though we use the same masked auto-encoding, our correspondence-aware decoder forces the model to learn better representations for point-tracking. We compare using a stride of 16 frames.

Method	Multi Frame	Kubric			DAVIS			Kinetics		
		AJ ↑	$\langle\delta^x_{\text{avg}}\rangle$ ↑	OA ↑	AJ ↑	$\langle\delta^x_{\text{avg}}\rangle$ ↑	OA ↑	AJ ↑	$\langle\delta^x_{\text{avg}}\rangle$ ↑	OA ↑
Supervised										
RAFT-C [7]		41.2	58.2	86.4	30.7	46.6	80.2	31.7	51.7	84.3
Kubric-VFS-Like [53]		51.9	69.8	84.6	33.1	48.5	79.4	40.5	59.0	80.0
RAFT-D [7]		61.8	79.1	87.9	34.1	48.9	76.1	72.1	**85.1**	92.1
COTR [54]		40.1	60.7	78.6	35.4	51.3	80.2	19.0	38.8	57.4
TAP-Net [55]		65.4	77.7	93.0	38.4	53.1	82.3	46.6	60.9	85.0
PIPs [56]	✓	59.1	74.8	88.6	42.0	59.4	82.1	35.3	54.8	77.4
Self-supervised										
CRW-C [36]		31.4	48.1	76.3	7.7	13.5	72.9	20.2	33.6	70.6
CRW-D [36]		35.8	52.4	80.9	23.6	38.0	77.2	21.9	36.8	70.4
DIFT-C [38]		28.3	45.2	69.0	18.1	33.0	68.8	19.8	33.7	68.7
DIFT-D [38]		41.6	59.8	83.9	29.7	48.2	77.2	19.5	34.4	70.1
Flow-Walk-C [52]		49.4	66.7	82.7	35.2	51.4	80.6	40.9	55.5	84.5
Flow-Walk-D [52]		51.1	68.1	80.3	24.4	40.9	76.5	46.9	65.9	81.8
ARFlow-C [57]		52.3	68.1	81.4	35.0	51.8	79.7	27.3	44.3	79.5
GMRW-C [37]		54.2	72.4	82.6	41.8	60.9	78.3	31.9	52.3	72.9
GMRW-D [37]		51.4	71.7	83.9	30.3	49.4	77.3	36.3	59.2	71.0
Video-GMAE zeroshot	✓	40.8	53.7	–	32.7	43.5	–	53.8	63.3	–
Video-GMAE base frozen	✓	60.8	70.7	96.6	45.2	55.2	91.4	61.7	68.9	97.1
Video-GMAE base finetune	✓	73.6	82.3	97.5	55.7	66.1	92.1	75.0	81.7	97.7
Video-GMAE large frozen	✓	62.4	71.9	96.6	46.7	55.8	91.8	65.1	72.0	97.4
Video-GMAE large finetune	✓	**74.0**	**82.4**	**97.6**	**57.9**	**67.7**	**93.5**	**75.1**	81.6	**97.9**

Table 2: Performance comparison on three video datasets. "Multi-Frame" indicates methods that jointly use multiple frames. *Video-GMAE* zero-shot does not predict occlusions, so we omit OA numbers for it. *Video-GMAE* shows strong performance across multiple datasets and multiple metrics. To compare with all the models, we use a stride of five for evaluations.

7.3 Comparison with Tracking Baselines

We compare *Video-GMAE* with state-of-the-art self-supervised methods like CRW [36], DIFT [38], and GMRW [37] and supervised tracking methods like RAFT [7] and TAP-Net [55]. We show results in Table 2, and Figure 6 shows some qualitative results of the fine-tuned point tracking. *Video-GMAE* outperforms the baselines at all model scales on all datasets. To compare with other models, we use a stride of five for evaluations [2]. We train two models, *Video-GMAE*-base and *Video-GMAE*-large. For each model, we train with a frozen encoder and a fully fine-tuned encoder. Among frozen encoder evaluations, we find that the large models outperform the base models. We see a similar scaling trend with the fine-tuned models, but overall, the fine-tuned models outperform the frozen encoder models. Our pre-training is fully self-supervised. However, to be comparable with other methods, we train a small cross-attention readout with supervised data. To have a fair comparison, in Table 2, we compare our method against both supervised and self-supervised approaches.

7.4 Frame Length Scaling

In this experiment, we investigate how our representations evolve as we train on longer frame sequences. We perform this ablation with *Video-GMAE*-base on frame lengths { 2, 4, 8, 16, 24 }. We evaluate these models on the strided TAPVid evaluation protocol, once with the stride equal to the number of frames the model was trained on, and once with a stride of two for a faithful comparison across the different ablations. We evaluate these using AJ on TAP-Vid Davis.

We find that training on longer frame sequences and evaluating on longer stride lengths leads to decreased AJ numbers. This is likely due to longer frame lengths inducing a stronger regularization from the correspondence-aware pre-training, which limits the quality of learned representations. When comparing on a fixed stride of two, we find that information of the future only helps with tracking to a certain extent. Having four frames and eight frames of information seems to help the model predict at a stride of two, but increasing it to 16 and 24 seems to decrease AJ.

Experiment	DAVIS (AJ) ↑
mean=True, rgb=True	44.7
mean=True, rgb=False	44.4
mean=False, rgb=True	42.5
mean=False, rgb=False	39.1

Table 3: Ablation study on the effects of integrating mean and RGB deltas during pretraining.

# Frames	Stride = # Frames ↑	Fixed Stride ↑
2	55.4	55.4
4	49.2	64.2
8	41.5	63.5
16	47.8	57.5
24	24.7	52.4

Table 4: Impact of the number of frames on TAP-Vid Davis AJ for *Video-GMAE*-base with varied strides and a fixed stride of two.

7.5 Delta Gaussian Ablations

We also explore the importance of the delta Gaussian parameters in modeling temporal evolution. Using *Video-GMAE*-large, we ablate three variants: (1) integrating only the mean components Δp_{ij}, (2) integrating only the RGB components Δr_{ij}, and (3) integrating neither of them, and having static Gaussians. This helps isolate the contribution of motion versus appearance changes in temporal modeling. We evaluate AJ on TAP-Vid Davis. We find that integrating mean correspondence not only enables zero-shot point-tracking but also improves the learned encoder latents for fine-tuned point-tracking.

8 Limitations

One of the main limitations of our work is the modeling of the camera. Throughout the pre-training, we assume a static camera, but this is not an ideal assumption when pretraining on internet-style videos. Because of the assumption about the static camera, we also fail to recover any metric 3-D information in the 3-D Gaussians. Another assumption we made in this work is regarding the correspondence-based regularization. While this is a weaker assumption than the prior one, and works reasonably well on short videos, when the number of frames at pre-training becomes very large, this regularization starts to hurt the learning. Additionally, during the pretraining, we only use 256 Gaussians per frame, which significantly limits the quality of our renderings. Finally, our zero-shot tracking approach does not predict occlusions. Since we are directly rendering the 3-D displacement lines into the 2-D image plane, we do not predict any occlusion scores for each point.

9 Societal Impact

Our work introduces a self-supervised approach to learning point correspondence in videos by predicting 3D Gaussian trajectories, enabling robust zero-shot tracking without relying on human annotations. This has positive societal implications by reducing the dependency on costly, labor-intensive labeled data, which can democratize access to high-quality video understanding models in domains such as robotics, assistive technologies, and environmental monitoring. However, as with any tracking technology, it also presents potential risks related to privacy and surveillance, especially if misused in contexts lacking consent or oversight. While our model does not include person identification and assumes static cameras during training, we recognize the broader ethical implications and emphasize the importance of responsible deployment, transparency, and alignment with legal and ethical norms in real-world applications.

10 Conclusion

In this paper, we introduced *Video-GMAE* for self-supervised learning from videos with built-in correspondence. By predicting the small changes in the Gaussian primitives over time, we enforced correspondence in videos. This also shows an alternative to patch-based video pretraining approaches. With this approach, we were able to pre-train self-supervised video models on large-scale datasets. Because of the correspondence-aware pretraining, our models show zero-shot capabilities in *any-point tracking*. Once fine-tuned, our models show very strong tracking performance across multiple datasets and multiple metrics. In summary, in this paper, we proposed a self-supervised video pretraining approach, which exhibits strong tracking performance on zero-shot and after finetuning.

References

[1] J. Piaget, *The construction of reality in the child*. Routledge, 1954. 1

[2] C. Doersch, A. Gupta, L. Markeeva, A. Recasens, L. Smaira, Y. Aytar, J. Carreira, A. Zisserman, and Y. Yang, "Tap-vid: A benchmark for tracking any point in a video," *Advances in Neural Information Processing Systems*, vol. 35, pp. 13610–13626, 2022. 1, 3, 5, 6, 8

[3] G. Le Moing, J. Ponce, and C. Schmid, "Dense optical tracking: Connecting the dots," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19187–19197, 2024.

[4] N. Karaev, I. Rocco, B. Graham, N. Neverova, A. Vedaldi, and C. Rupprecht, "Cotracker: It is better to track together," in *European Conference on Computer Vision*, pp. 18–35, Springer, 2024. 1

[5] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, *et al.*, "Sam 2: Segment anything in images and videos," *arXiv preprint arXiv:2408.00714*, 2024. 1

[6] P. Dendorfer, A. Osep, A. Milan, K. Schindler, D. Cremers, I. Reid, S. Roth, and L. Leal-Taixé, "Motchallenge: A benchmark for single-camera multiple target tracking," *International Journal of Computer Vision*, vol. 129, pp. 845–881, 2021. 1

[7] Z. Teed and J. Deng, "Raft: Recurrent all-pairs field transforms for optical flow," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pp. 402–419, Springer, 2020. 1, 3, 8

[8] A. Jabri, A. Owens, and A. Efros, "Space-time correspondence as a contrastive random walk," *Advances in neural information processing systems*, vol. 33, pp. 19545–19560, 2020. 1

[9] A. Shrivastava and A. Owens, "Self-supervised any-point tracking by contrastive random walks," in *European Conference on Computer Vision*, pp. 267–284, Springer, 2024.

[10] S. Stojanov, D. Wendt, S. Kim, R. Venkatesh, K. Feigelis, J. Wu, and D. L. Yamins, "Self-supervised learning of motion concepts by optimizing counterfactuals," *arXiv preprint arXiv:2503.19953*, 2025. 1

[11] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022. 1, 2

[12] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," *ACM Transactions on Graphics*, vol. 42, no. 4, pp. 1–14, 2023. 1, 3

[13] Z. Tong, Y. Song, J. Wang, and L. Wang, "Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training," 2022. 2, 3, 6, 7

[14] C. Feichtenhofer, Y. Li, K. He, *et al.*, "Masked autoencoders as spatiotemporal learners," *Advances in neural information processing systems*, vol. 35, pp. 35946–35958, 2022. 2, 4, 7

[15] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3733–3742, 2018. 2

[16] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*, pp. 1597–1607, PMLR, 2020. 2

[17] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020. 2

[18] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9650–9660, 2021. 2

[19] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018. 2

[20] H. Bao, L. Dong, S. Piao, and F. Wei, "Beit: Bert pre-training of image transformers," *arXiv preprint arXiv:2106.08254*, 2021. 2

[21] Y. Li, H. Mao, R. Girshick, and K. He, "Exploring plain vision transformer backbones for object detection," in *European Conference on Computer Vision*, pp. 280–296, Springer, 2022. 2

[22] Y. Xu, J. Zhang, Q. Zhang, and D. Tao, "ViTPose: Simple vision transformer baselines for human pose estimation," in *Advances in Neural Information Processing Systems*, 2022. 2

[23] I. Radosavovic, T. Xiao, S. James, P. Abbeel, J. Malik, and T. Darrell, "Real-world robot learning with masked visual pre-training," in *Conference on Robot Learning*, 2022. 2

[24] L. Wang, B. Huang, Z. Zhao, Z. Tong, Y. He, Y. Wang, Y. Wang, and Y. Qiao, "Videomae v2: Scaling video masked autoencoders with dual masking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14549–14560, 2023. 2

[25] J. Rajasegaran, I. Radosavovic, R. Ravishankar, Y. Gandelsman, C. Feichtenhofer, and J. Malik, "An empirical study of autoregressive pre-training from videos," *arXiv preprint arXiv:2501.05453*, 2025. 2

[26] S. Ren, Q. Yu, J. He, X. Shen, A. Yuille, and L.-C. Chen, "Beyond next-token: Next-x prediction for autoregressive visual generation," *arXiv preprint arXiv:2502.20388*, 2025. 2

[27] P. Tong, E. Brown, P. Wu, S. Woo, A. J. V. IYER, S. C. Akula, S. Yang, J. Yang, M. Middepogu, Z. Wang, *et al.*, "Cambrian-1: A fully open, vision-centric exploration of multimodal llms," *Advances in Neural Information Processing Systems*, vol. 37, pp. 87310–87356, 2024. 2

[28] E. Fini, M. Shukor, X. Li, P. Dufter, M. Klein, D. Haldimann, S. Aitharaju, V. G. T. da Costa, L. Béthune, Z. Gan, *et al.*, "Multimodal autoregressive pre-training of large vision encoders," *arXiv preprint arXiv:2411.14402*, 2024. 2

[29] S. Liu, T. Li, W. Chen, and H. Li, "Soft rasterizer: A differentiable renderer for image-based 3d reasoning," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 7708–7717, 2019. 3

[30] C. Lassner and M. Zollhöfer, "Pulsar: Efficient sphere-based neural rendering," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021. 3

[31] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021. 3

[32] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman, "Mip-nerf 360: Unbounded anti-aliased neural radiance fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5470–5479, 2022. 3

[33] M. Levoy, "Efficient ray tracing of volume data," *ACM Transactions on Graphics (ToG)*, vol. 9, no. 3, pp. 245–261, 1990. 3

[34] C. Tomasi and T. Kanade, "Detection and tracking of point," *Int J Comput Vis*, vol. 9, no. 137-154, p. 3, 1991. 3

[35] C. Doersch, Y. Yang, M. Vecerik, and et al., "Tapir: Tracking any point with per-frame initialization and temporal refinement," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 3

[36] A. Jabri, A. Owens, and A. A. Efros, "Unsupervised learning of visual representations by dense cycle-consistency," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 3, 8

[37] A. Shrivastava and A. Owens, "Self-supervised any-point tracking by contrastive random walks," in *European Conference on Computer Vision (ECCV)*, 2024. 3, 5, 6, 8

[38] L. Tang, M. Jia, Q. Wang, C. P. Phoo, and B. Hariharan, "Emergent correspondence from image diffusion," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 3, 6, 8

[39] S. Koppula, I. Rocco, Y. Yang, J. Heyward, J. Carreira, A. Zisserman, G. Brostow, and C. Doersch, "Tapvid-3d: A benchmark for tracking any point in 3d," *arXiv preprint arXiv:2407.05921*, 2024. 3

[40] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018. 3

[41] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2023. 3

[42] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," 2021. 3

[43] H. Bao, L. Dong, S. Piao, and F. Wei, "Beit: Bert pre-training of image transformers," 2022. 3

[44] C. Feichtenhofer, H. Fan, Y. Li, and K. He, "Masked autoencoders as spatiotemporal learners," 2022. 3, 6

[45] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," 2021. 3

[46] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," *ACM Transactions on Graphics (TOG)*, vol. 42, no. 4, pp. 1–14, 2023. 3, 5

[47] J. Rajasegaran, X. Chen, R. Li, C. Feichtenhofer, J. Malik, and S. Ginosar, "Gaussian masked autoencoders," *arXiv preprint arXiv:2501.03229*, 2025. 3, 4

[48] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, *et al.*, "The kinetics human action video dataset," *arXiv preprint arXiv:1705.06950*, 2017. 4

[49] K. Greff, F. Belletti, L. Beyer, C. Doersch, Y. Du, D. Duckworth, D. J. Fleet, D. Gnanapragasam, F. Golemo, C. Herrmann, *et al.*, "Kubric: A scalable dataset generator," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3749–3761, 2022. 5, 6, 7

[50] J. Carreira, D. Gokay, M. King, C. Zhang, I. Rocco, A. Mahendran, T. A. Keck, J. Heyward, S. Koppula, E. Pot, *et al.*, "Scaling 4d representations," *arXiv preprint arXiv:2412.15212*, 2024. 5

[51] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," 2023. 5

[52] Z. Bian, A. Jabri, A. A. Efros, and A. Owens, "Learning pixel trajectories with multiscale contrastive random walks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. Model referred to as *Flow-Walk*. 6, 8

[53] K. Greff, F. Belletti, L. Beyer, and et al., "Kubric: A scalable dataset generator," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. CVPR Dataset&Benchmark track. 8

[54] W. Jiang, E. Trulls, J. Hosang, A. Tagliasacchi, and K. M. Yi, "Cotr: Correspondence transformer for matching across images," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 8

[55] C. Doersch, A. Gupta, L. Markeeva, and et al., "Tap-net: Tracking any point via cost-volume matching," in *NeurIPS Datasets & Benchmarks (TAP-Vid paper that introduces TAP-Net baseline)*, 2022. 8

[56] A. W. Harley, Z. Fang, and K. Fragkiadaki, "Particle video revisited: Tracking through occlusions using point trajectories," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 8

[57] L. Liu, J. Zhang, R. He, Y. Liu, Y. Wang, and et al., "Learning by analogy: Reliable supervision from transformations for unsupervised optical flow estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 8

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We ran evals on multipe datasets to confirm our results.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of our work in the main paper.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: No theoretical results in the paper.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We have included all the hyper-parameters for pretraining and finetuning of our models in the appendix, and we will also release our models and code.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We will release the code and the models after publication, and no new datasets were used in the paper.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: Yes, we use standard datasets for evaluation and training, and we use the standard splits. All the hyper-parameters are discussed in the appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: We follow the standard practice in these datasets for evaluation.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We discussed the number of gpus used for training and finetuning our models.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Yes, we follow the ethics guidelines and our research aligns with it.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the social impact of our work in the main paper.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: We do not release any language or image generative models.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: yes, all the creators of the dataset are properly cited and credited.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: yes, we will release the full documentation on how to run the code as how to use our models, after review period.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: no human subjects were involved in the paper.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: no human subjects were involved in the paper.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [No]

Justification: No llms were used in the writing, editing or formatting.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.