CEFFYL: MODELLING DEFORMABLE OBJECT SHAPE FOR IMPROVED POINT TRACKING

Anonymous authors
Paper under double-blind review

ABSTRACT

Point tracking algorithms have consistently struggled with long-term and complex motion of objects. We introduce CEFFYL, a shape modelling method that utilises **C**anonical shape **E**mbeddings and **F**eature **F**iltering to **Y**ield **L**ong-term point tracks. Specifically, we model the amodal representation of each pixel as a set of points on the rest pose of the object category. Once this mapping is established, correspondence and point tracking is estimated from this shared canonical space. We additionally learn a richer descriptor feature for local correspondences within the canonical space. We train CEFFYL on synthetic data of horses, particularly difficult objects to track due to their deformable nature and complex leg motion at speed. CEFFYL outperforms state-of-the-art point tracker AllTracker by 8.9 and 16.7 points on Horse10 and long Showground videos, respectively. We also outperform the single-frame DualPM shape model, the most recent work of this type, increasing performance by 7.9 and 8.5 points on the same metrics.

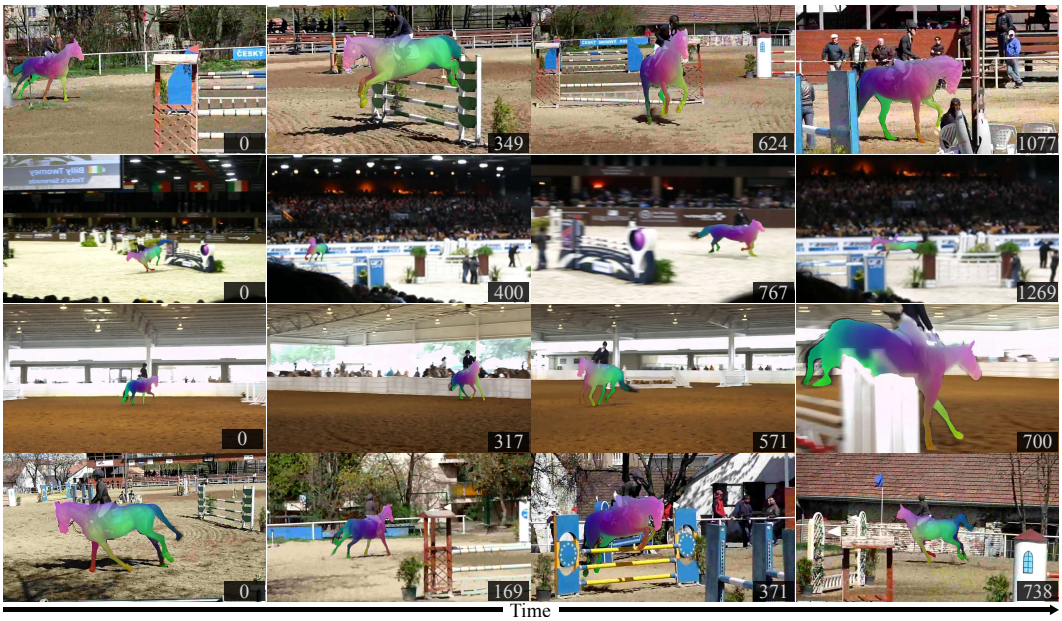


Figure 1: CEFFYL maps each pixel of the horse to a canonical shape, along with predictions for self-occluded pixels via an amodal pointmap representation. We show four examples from 'Showground' benchmark - dense correspondence is robust over long time and large pose variations.

1 INTRODUCTION

The ability to track arbitrary points across the frames of a video has garnered a lot of attention in recent years. Countless methods have been explored, including independent tracking (Harley et al., 2022; Zheng et al., 2023), joint tracking (Karaev et al., 2024), pseudo-label training (Doersch et al., 2024; Karaev et al., 2025), all-pixel deformations (Harley et al., 2025), tracking with 3D priors (Xiao et al., 2024; Ngo et al., 2025) and test-time optimisation of features on the target video (Tumanyan et al., 2024). However, these prior works are built upon a straight-forward principle: a query point

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

allows for the extraction of a query feature, a cost volume is constructed contrasting this query feature and the features of one or more target frames and then the position is regressed by a trained network. These methods do not seek to model or understand the objects that the points lie on or are a part of. Instead, they rely on the strength of the features themselves. Despite strong generalisation capabilities on a number of popular benchmarks (Doersch et al., 2022; Vecerik et al., 2024), these methods are known to struggle to maintain robust performance for long-term and complex motion (Zheng et al., 2023; Guerrier et al., 2026; Doersch et al., 2022; Xiao et al., 2024).

In contrast, recent developments in object, scene and motion modelling open a potentially new avenue for tackling these perennial problems of long-term point tracking. Although powerful optimisation-based methods have been the norm for many years (Agarwal et al., 2011; Schonberger & Frahm, 2016; Mur-Artal et al., 2015; Engel et al., 2014), more recent deep-learning approaches have focused the task of modelling shape on solutions that rely on large, feed-forward models (Wang et al., 2024a; Leroy et al., 2024; Wang et al., 2025a; 2026). The common output format used by these models is the pointmap. This is a representation that maps every pixel to a coordinate in a 3D space.

Instead of attempting *general point tracking*, we instead train a *specialised shape model for an object category*, which allows us to track points on instances of that category robustly and for very long. Specifically, we propose to re-formulate point tracking as a task of modelling an object's canonical amodal shape. Once the shape can be understood consistently across frames, correspondence and stable long-term point tracking can be estimated. To be specific, we introduce a novel framework named CEFFYL[1], which stands for **C**anonical shape **E**mbeddings and **F**eature **F**iltering to **Y**ield **L**ong-term point tracks. Our work is inspired by the work of Kaye et al. (2025), which highlights the efficacy of modelling objects with amodal pointmaps and Guerrier et al. (2026), which makes a case for tracking via point-to-point correspondence matching. CEFFYL is a spatio-temporal transformer-based network that takes as input two object-centric frames: a query and a target. The model then returns a set of amodal pointmaps, mapping every query and target pixel to a shared canonical space whilst also capturing the self-occluded regions of the object. This representation then accommodates object-centric point tracking through nearest-neighbour search. Our contributions are as follows:

1. We re-formulate object-centric point tracking as a task of modelling the canonical shape, whereby correspondence is estimated through neighbourhood search within this representation.
2. We introduce CEFFYL, a transformer-based network that models the amodal mapping of object pixels to a shared canonical space facilitating correspondence and long-term tracking.
3. We showcase that learning rich descriptor features per amodal pixel improves downstream tracking performance through local refinement.
4. We provide a new benchmark of sparsely annotated set of 9 long challenging horse showground videos for point tracking, which we call "Showground".

2 Related Works

Optical Flow & Point Tracking. Early optical flow approaches were optimisation based (Horn & Schunck, 1981; Lucas & Kanade, 1981), whereas modern approaches rely on deep learning, combining learnt features with iterative refinement. This paradigm was popularised by RAFT (Teed & Deng, 2020), which builds feature cost volumes before performing iterative refinement. Subsequent works improve performance through better initialisation and uncertainty modelling (Wang et al., 2024b), transformer-based approaches (Huang et al., 2022; Shi et al., 2023), and strong vision priors from foundation models (Zhang et al., 2026). Cost volumes and iterative refinement have also been introduced to point tracking (Harley et al., 2022). Recent works have demonstrated improvements through joint tracking of queries (Karaev et al., 2024; 2025), pseudo-labelling of real data for fine-tuning (Karaev et al., 2025; Doersch et al., 2024) and tracking with 3D priors (Xiao et al., 2024; Ngo et al., 2025). More recently, AllTracker (Harley et al., 2025) achieved state-of-the-art performance by unifying optical flow and point tracking to predict all-pixel, multi-frame trajectories.. Both modern optical flow and point tracking methods rely on large amounts of synthetic training data for supervision (e.g. Greff et al., 2022; Zheng et al., 2023; Wang et al., 2020; Mehl et al., 2023). This is because labelling points and flow in real videos is difficult and time-consuming, and the real data that does exist is narrow in domain: KITTI (Geiger et al., 2013), for instance, provides semi-dense ground truth for rigid street scenes, leaving deformable motion unlabelled. However, a

[1]Ceffyl is Welsh for "horse" (pronounced keff-ill), as our main training and evaluations are for this animal.

number of human-annotated real-world point tracking benchmarks have helped confirm the efficacy of the models (Doersch et al., 2022; Vecerik et al., 2024; Darkhalil et al., 2025; Mathis et al., 2021).

Promptable Video Object Segmentation. Segmenting an object throughout a video is another form of tracking, at the level of regions rather than points. Earlier successful models focused on caching features from previous frame masks to inform new predictions (Oh et al., 2019; Cheng et al., 2021), improving multi-object segmentation with transformers (Yang et al., 2021) and compressing past frames into a compact, multi-store memory for long-term propagation (Cheng & Schwing, 2022). The SAM2 (Ravi et al., 2025) and SAM3 (Carion et al., 2026) video models extended the possible prompt inputs to points, bounding boxes and text, creating foundation video object segmentation models, capable of reliably segmenting and tracking arbitrary objects over long video lengths.

Pixel-Level Correspondence. Similar to our work, pixel-level correspondence is concerned with the pixel matches between two images. Early works utilise multi-scale CNN features, transformer architectures and matching refinements (Min et al., 2019; Liu et al., 2020; Cho et al., 2022). More recently, features learnt by self-supervised foundation models (Oquab et al., 2024) and Stable Diffusion models (Zhang et al., 2023) have been shown to exhibit strong correspondence understanding. This has led to more performant correspondence models that train aggregation networks on top of fused SD-DINO (Zhang et al., 2023) features (Zhang et al., 2024), adapters on frozen DINOv2 (Oquab et al., 2024) backbones (Cuttano et al., 2026) and test-time optimisation refinement models on top of DINOv2 features (Tumanyan et al., 2024).

Feed-forward 3D/4D Reconstruction for Scenes and Objects. Modern feed-forward 3D reconstruction models trained on large datasets have shown impressive capabilities in representing static scenes (Wang et al., 2025a; 2026; Leroy et al., 2024; Wang et al., 2024a). These have been extended to dynamic reconstruction, such as joint temporal transformations and single-frame reconstructions (Feng et al., 2025; Sucar et al., 2025; 2026), aggregation of frames with internal states (Wang et al., 2025b; Guédon et al., 2026) and explicit motion estimation (Jiang et al., 2026; Luo et al., 2026). These methods share the pointmap representation popularised by DUSt3R (Wang et al., 2024a). However, this representation does not fully capture the extent of unseen geometry, which limits object modelling. Object models initially focused on mesh-based approaches (Kulkarni et al., 2020; Wu et al., 2023; Jakab et al., 2024; Li et al., 2024b), but more recent work recovers amodal object geometry (whereby self-occluded pixels are also predicted), either generatively (Wu et al., 2025) or by extending the pointmap (Kaye et al., 2025). Specifically, DualPM (Kaye et al., 2025) maps each pixel to both a posed and a canonical space, modelling deformation as well as correspondence.

Our work builds on shape modelling, utilising it for long-term pixel-level correspondence. By modelling an object's shape, we not only achieve long-term tracking but can also accurately track occluded parts of the object in every frame. We describe our method next.

3 METHOD

Our method takes two images as input and maps the position of each pixel on an object to a shared amodally-complete canonical shape. Once this mapping has been made, rough correspondence is achieved through nearest-neighbour search. We explain the amodal pointmap in Section 3.1, the shared canonical space in Section 3.2 and how to extract point tracks from this space in Section 3.3. Our architecture and training losses are then explained in Section 3.4.

3.1 PRELIMINARIES: AMODAL POINTMAPS

The amodal pointmap, as introduced in Kaye et al. (2025), is an extension of the earlier pointmap representation used in many recent 3D and 4D reconstruction works (Wang et al., 2024a; Leroy et al., 2024; Wang et al., 2025a). Whereas a pointmap $Y \in \mathbb{R}^{H \times W \times 3}$ provides a mapping between each image pixel and a 3D point, an amodal pointmap seeks to also represent the self-occluded points that make up the unseen part of the object or scene in question. More formally, an amodal pointmap $X \in \mathbb{R}^{H \times W \times A \times 3}$ stores every point at which the pixel's ray intersects with the surface of the object. A is the number of amodal layers, which can be tuned to whatever is reasonable for the object in question, and is an even number, as every entry into the object's mesh must have an exit. This is a particularly powerful representation, as it leverages the convenience of the image grid while also allowing for a complete representation of the object's 3D shape. Some points in the

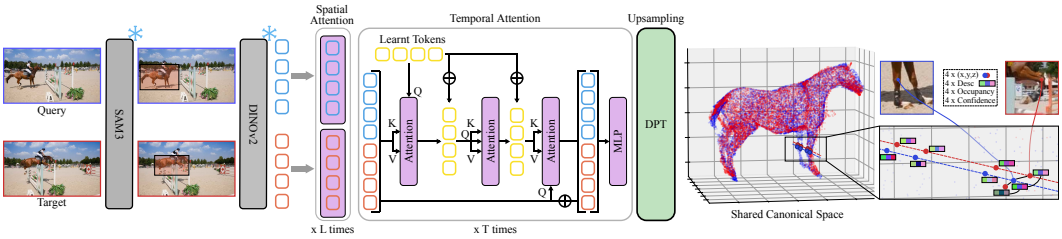


Figure 2: **CEFFYL architecture overview.** DINOv2 (Oquab et al., 2024) features are extracted from SAM3 (Carion et al., 2026) generated masks. Per-frame spatial attention is calculated before these features read-write information between learned temporal aggregation tokens. Lastly, DPT (Ranftl et al., 2021) heads upsample features back to original image resolution and extract shared canonical pointmaps, occupancy volumes and descriptors.

image may intersect with the object less than A times or not at all. For this reason, many cells of X are empty. We therefore pair X with a binary occupancy map $O \in \{0,1\}^{H\times W\times A}$, as in Kaye et al. (2025), that marks, for each pixel, which layers have a valid intersection.

3.2 Shared Canonical Space

Given a collection of S images $I \in \mathbb{R}^{S\times H\times W\times 3}$ depicting an object in different poses or articulations, the *canonical shape* is the representation of this object in a single, standardised pose. In line with prior work, we choose the neutral (or rest) pose as canonical. We learn a mapping from images to this space $G(I^i) \to X \in \mathbb{R}^{H\times W\times A\times 3}$, where X is an amodal pointmap. Unlike Kaye et al. (2025), where this mapping is learnt from a single image, in our work, we propose to train this mapping from a pair of frames (a "query" and a "target") to a shared canonical space. This joint formulation offers two key advantages. Firstly, observing the object from two distinct viewpoints better constrains its overall 3D shape, recovering structural geometry that is inherently ambiguous in a single-frame projection. Secondly, it directly facilitates point tracking by resolving occlusion ambiguities: the query frame acts as a visual anchor, safeguarding the context and facilitating robust correspondence even when those points become occluded in the target frame. Specifically, $G(I^Q, I^T) \to X^Q, X^T \in \mathbb{R}^{H\times W\times A\times 3}$ jointly maps the query (I^Q) and target (I^T) to a pair of amodal pointmaps in the shared canonical space. We estimate point tracks from this space next.

3.3 Point Tracking in the Canonical Space

Given two images, $I^Q, I^T \in \mathbb{R}^{H\times W\times 3}$ and their respective amodal pointmaps $X^Q, X^T \in \mathbb{R}^{H\times W\times A\times 3}$, we can perform a correspondence task using nearest-neighbour search. Specifically, given a query pixel coordinate $p^Q \in \mathbb{R}^2$ in I^Q, its canonical point is $x^Q = X^Q[p^Q, 0] \in \mathbb{R}^3$, taken from the zeroth amodal layer, representing the visible surface. We find the closest point to x^Q in the target frame's amodal pointmap, searching over all of the target frame's pixel coordinates and amodal layers where the predicted occupancy O^T is above some threshold γ:

$$(p^T, a^T) = \underset{(p,\,a)\,:\,O^T[p,a]>\gamma}{\arg\min} \left\| x^Q - X^T[p,a] \right\|_2 \tag{1}$$

This gives the coordinate p^T locating the query in the target frame, and the amodal layer a^T at which it is found. If $a^T > 0$, the point is believed to be occluded in the target frame. To estimate a trajectory of the point, we find p^T for all frames T in the sequence, by independently pairing the query frame with each other frame (i.e., $T-1$ forward passes and searches).

Local refinement. Nearest-neighbour in the 3D canonical space can sometimes yield locally noisy correspondences. To mitigate this, rather than strictly selecting the closest point, we retrieve the K nearest neighbours in the canonical space, and find the point in this neighbourhood with the highest cosine similarity to the query, using learned high-dimensional descriptors (detailed in Section 3.4).

3.4 Architecture

The architecture of CEFFYL can be seen in Figure 2. Our model takes as input a query image and a target image, $I^Q, I^T \in \mathbb{R}^{H\times W\times 3}$. Firstly, masks $m^Q, m^T \in [0,1]^{H\times W}$ are predicted using

SAM3 (Carion et al., 2026) to localise the object in each image. In the query image, we use point queries as prompts for SAM3 and propagate the mask. We use a DINOv2 (Oquab et al., 2024) image backbone to extract features within the object bounding box: $P^Q, P^T \in \mathbb{R}^{N \times E}$, where N is the number of patches and E is the feature dimension.

Spatial Attention. Our spatial attention blocks process each frame independently, with L self-attention transformer blocks. We use RoPE (Su et al., 2024) position embeddings.

Temporal Attention. After L layers of spatial attention, we share information across the query/target pair, processing each layer's token set independently. Unlike previous works that use global attention for cross-frame context, we learn a set of aggregation tokens, which cross-attend to the patch tokens, self-attend, and then broadcast updates to the patch tokens. Specifically, we begin by adding learnt query/target indicator embeddings $e_Q, e_T \in \mathbb{R}^E$ to each frame's tokens:

$$\mathbf{Z} = \begin{bmatrix} \mathbf{P}^Q + \mathbf{e}_q \\ \mathbf{P}^T + \mathbf{e}_t \end{bmatrix} \in \mathbb{R}^{(2N) \times E} \tag{2}$$

Learnt aggregation tokens $M \in \mathbb{R}^{N_m \times E}$ then collect information from Z through cross-attention:

$$M \leftarrow M + Attn(M, Z, Z) \tag{3}$$

where $Attn$ denotes the standard multi-headed attention operation (with inputs being queries, keys and values). The aggregation tokens then share information using self attention:

$$M \leftarrow M + Attn(M, M, M) \tag{4}$$

The updated aggregated tokens then write information back through cross attention:

$$Z \leftarrow Z + Attn(Z, M, M) \tag{5}$$

followed by an MLP. The output is reshaped to $P^Q, P^T \in \mathbb{R}^{N \times E}$.

DPT Upsampling. The initial patchification of the input images results in a feature resolution far below that of the original image $N << (H \times W)$. Point tracking requires fine-grained localisation, so we follow previous reconstruction works (Wang et al., 2024a; Leroy et al., 2024; Wang et al., 2025a; 2026) and upsample using a Dense Prediction Transformer (DPT) head (Ranftl et al., 2021). The output of each temporal attention layer is fed into the DPT head where it is fused and upsampled into one, dense output. The DPT head is also responsible for outputting the confidence of each amodal point $C \in \mathbb{R}^{H \times W \times A}$, and occupancy $O \in \mathbb{R}^{H \times W \times A}$.

High-Dimensional Descriptors. We also propose a second DPT head which estimates amodal descriptor features for every pixel $D \in \mathbb{R}^{H \times W \times A \times F}$, where F is the dimension size. This is used for local refinement, where target candidates in the canonical space are re-ranked according to their cosine similarity to the query's descriptor feature.

Training Losses. Our method is trained with three losses. The first is a confidence-aware regression loss on the canonical pointmap outputs $X^Q, X^T \in \mathbb{R}^{H \times W \times A \times 3}$, supervised with the ground truth (for which details are provided in Section 4.1). Specifically, we use a smooth L1 loss on each pointmap. Following prior works Wang et al. (2024a); Leroy et al. (2024); Wang et al. (2025a; 2026) we wrap this in a confidence-aware weighting:

$$\mathcal{L}_{conf} = \sum_{i \in \{Q,T\}} \sum_{j \in O} C_j^i \overline{L1}(\hat{X}_j^i - X_j^i) - \alpha \log(C_j^i) \tag{6}$$

where $\overline{L1}$ is smooth L1, $i \in \{Q, T\}$ is the sample view of query or target, j is any pixel in the object's occupancy mask $O \in \mathbb{R}^{H \times W \times A}$ and $C \in \mathbb{R}^{H \times W \times A}$ is the predicted confidence.

The second loss is a binary cross-entropy on the occupancy mask O, denoted $\mathcal{L}_{occup}$. We supervise with an equal number of positive (on-object) and negative (off-object) samples.

The final loss is an infoNCE loss (van den Oord et al., 2018) used to train the descriptor head. For each pair of training frames, we sample a random pixel coordinate from the object's mask in the query frame. The canonical vertex for this point is used to find the "positive" 2D pixel

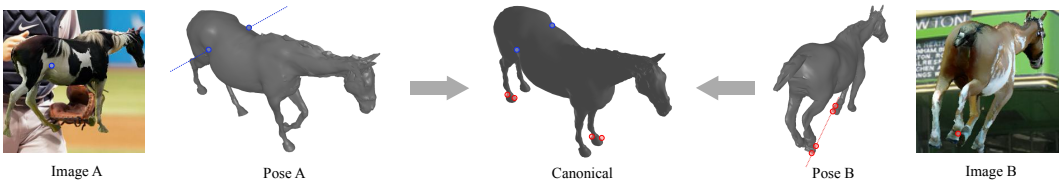


Figure 3: **Training data generation pipeline.** A ray is shot through each pixel into the posed mesh, the intersections are transformed back to the rest pose and then stored in the amodal representation.

correspondence in the target frame. The pixels directly surrounding this are used as hard negatives. Our descriptor loss is therefore:

$$\mathcal{L}_{desc} = -\log \frac{\exp(d_q^T d_t/\tau)}{\exp(d_q^T d_t/\tau) + \sum_{k=1}^{K} \exp(d_q^T d_{n_k}/\tau)} \tag{7}$$

where d_q, d_t denote the descriptors for the query and target correspondence and d_{n_k} denote the descriptors for the negative pixels.

Although all components of the model can be learnt together, we find that it is better to first train a model without a descriptor head using $\mathcal{L}_{stage1} = \mathcal{L}_{conf} + \mathcal{L}_{occup}$, and then freeze the network and only train the new DPT descriptor head using $\mathcal{L}_{stage2} = \mathcal{L}_{desc}$.

4 Experiments

4.1 Training Data and Implementation Details

We train CEFFYL using the VarenPoser dataset (Lyu et al., 2026), which contains 1171 object-centred clips of synthetic posed horses on random backgrounds, complete with shape, camera, and pose annotations. To generate ground-truth amodal pointmaps, we raycast from camera pixels to the mesh (see Figure 3). Each intersection is transformed into the canonical rest pose using the inverse pose parameters. For every populated pixel coordinate (x_i, y_i) at amodal layer a, we set the ground-truth occupancy $O[y_i, x_i, a] = 1$. Additionally, we record the exact vertex ID for each intersection to establish cross-frame pixel correspondences for training the descriptor head. To optimize data generation while maintaining sufficient supervision, we only raycast a random 10% of pixels every 5^{th} frame, yielding a final training set of 1160 sequences of 120 frames each.

The input images are cropped around the SAM3 predicted masks with a buffer of 10 pixels and resized to 518×518 resolution. For DINOv2, we utilise the pre-trained ViT-L. We use $L = 4$ frame spatial attention blocks as in Wang et al. (2025a; 2026). We use $T = 4$ temporal blocks, each with 3 multi-headed attention layers, 64 learnt tokens of dimension 1024, a two-layer MLP with a hidden dimension of 4096, GELU activation, and a linear projection back to the original dimension of 1024. All attention blocks use 8 attention heads. We use a batch size of 12, learning rate of $5e^{-5}$ over $100k$ training pairs for $50k$ steps, with a warmup of 500 steps and cosine annealing to $5e^{-6}$. For stage 2, we train for 2000 steps with a learning rate of $1e^{-4}$. Training takes 12 hours on one H100 GPU.

4.2 "Showground" and other Evaluation Datasets

We introduce a new benchmark for long-term tracking with pose variations. We select 9 publicly available videos from Wikimedia Commons, with creative commons licenses, of horses competing in indoor and outdoor show grounds. These videos range from 701 to 1961 frames per clip (compared to $\leq$ 300 frames in Horse10 dataset (Mathis et al., 2021) and $\leq$ 104 in TAP-Vid-DAVIS (Doersch et al., 2022)) and often depict horses galloping, turning quickly and jumping over barriers (see sample frames in Figure 1). We follow Darkhalil et al. (2025) by annotating a query frame at the start of each clip as well as a sparse set of 10 evaluation frames. There is an average of 7 points per clip, annotated around common body parts such as nose, ears, knee joints and tail, and these are hand annotated by a single annotator for consistency. We refer to this benchmark as "Showground".

Additionally, we evaluate on two established benchmarks: Horse10 (Mathis et al., 2021), which includes 30 videos of horses being led by a human, and has been used to evaluate point trackers

Table 1: Comparative results for d_{avg} on three benchmarks - Horse10, PFRED and our newly introduced long-term Showground benchmark.

Models		Horse10	PFERD	Showground
CoTracker3 (Karaev et al., 2025)		46.7	50.6	20.2
AllTracker (Harley et al., 2025)		49.0	55.3	44.1
PointSt3R (Guerrier et al., 2026)		47.6	51.5	30.6
DINOv2 NN (Oquab et al., 2024)		40.3	50.5	52.1
DINO-Tracker (Tumanyan et al., 2024)		42.5	51.6	37.0
GeoAware-SC (Zhang et al., 2024)		45.6	48.3	50.3
MARCO (Cuttano et al., 2026)		49.3	54.2	53.8
OmniX (Jiang et al., 2026)		19.7	35.8	0.2
St4RTrack (Feng et al., 2025)		12.2	40.7	9.0
CoTracker3 (Karaev et al., 2025)	+SAM3	48.9	67.7	48.2
AllTracker (Harley et al., 2025)	+SAM3	54.3	65.5	53.5
PointSt3R (Guerrier et al., 2026)	+SAM3	51.7	63.5	44.6
DINOv2 NN (Oquab et al., 2024)	+SAM3	45.6	69.7	65.9
DINO-Tracker (Tumanyan et al., 2024)	+SAM3	46.0	63.4	51.9
GeoAware-SC (Zhang et al., 2024)	+SAM3	48.0	64.1	57.8
MARCO (Cuttano et al., 2026)	+SAM3	54.8	73.3	64.5
OmniX (Jiang et al., 2026)	+SAM3	5.9	25.4	3.3
St4RTrack (Feng et al., 2025)	+SAM3	21.9	38.0	20.1
DualPM (Kaye et al., 2025)		55.3	76.8	61.7
CEFFYL		**63.2**	**84.0**	**70.2**



Figure 4: Qualitative examples of point tracks on Horse10 (left), PFERD (middle) and Showground (right). Coloured points are predictions. Yellow lines show distances to ground truth. The top row shows AllTracker using the SAM3 detector; the bottom row shows CEFFYL.

(e.g., Sun et al. (2024); Harley et al. (2025)). The second benchmark is the PFERD (Li et al., 2024a) dataset, with videos of horses fitted with roughly 130 motion capture markers, providing us with both pose parameters and 2D point tracks, but unfortunately no visibility annotations. We estimate visibility by calculating self-occlusion from the depth of the predicted pose mesh and occlusion by the person leading the horse. In total, we pick 10 240-frame clips from across 3 PFERD horses.

As is consistent with point tracking works (Doersch et al., 2022; Harley et al., 2025), we primarily evaluate models with the d_{avg} metric. This measures the average percentage of points which fall within pixel thresholds of $[1, 2, 4, 8, 16]$ after normalising resolution to 256×256.

4.3 Point Tracking Results

In Table 1, we compare CEFFYL to state-of-the-art baselines across three paradigms: point tracking (Harley et al., 2025; Karaev et al., 2025), correspondence (Tumanyan et al., 2024; Oquab et al., 2024; Zhang et al., 2024; Cuttano et al., 2026), and dynamic reconstruction (Guerrier et al., 2026; Jiang et al., 2026; Feng et al., 2025), as well as our direct precursor, DualPM (Kaye et al., 2025).

As CEFFYL and DualPM rely on object masks, we evaluate all baselines using both full images and objects localized via SAM3. Constraining the search space predictably improves traditional methods: AllTracker, CoTracker3, and PointSt3R gain up to $+5.3$, $+2.2$, and $+4.1$ points on Horse10, and $+9.4$, $+28.0$, and $+14.0$ on Showground. However, CEFFYL still outperforms the strongest tracker variant (AllTracker+SAM3) by $+8.9$, $+18.5$, and $+16.7$ points on Horse10, PFERD, and Showground, respectively. PFERD proves especially difficult for traditional trackers, as markers on the object boundary frequently slip onto the background. Against 2D correspondence models, CEFFYL outperforms the strongest baseline, MARCO (Cuttano et al., 2026), by $+8.4$, $+10.7$, and $+5.7$ points on Horse10, PFERD, and Showground. Since these baselines also rely on DINOv2

Table 2: Occlusion accuracy (OA) and Average Jaccard (AJ) for our three benchmarks.

	Horse10		PFERD		Showground	
Model	OA	AJ	OA	AJ	OA	AJ
AllTracker	77.7	36.2	61.3	28.9	61.4	32.1
AllTracker + SAM3	72.6	38.1	63.6	28.6	36.8	19.4
CEFFYL	**87.7**	**49.4**	**75.3**	**47.1**	**85.1**	**53.5**

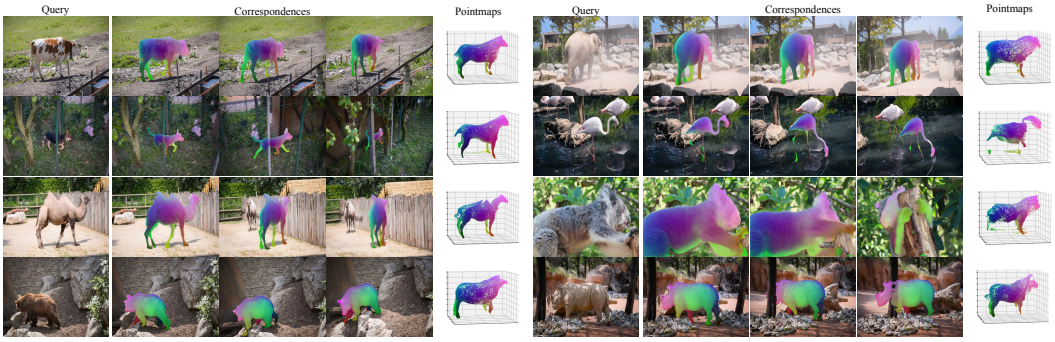


Figure 6: Zero-shot generalisation of CEFFYL (trained only on horses) to non-horse animals.

features, this margin isolates the impact of CEFFYL's design, demonstrating that our performance gains stem from explicit 3D shape modelling rather than feature representations alone.

Dynamic 4D reconstruction methods underperform point trackers and correspondence models on these benchmarks. Interestingly, their accuracy drops when combined with SAM3, likely because they rely on full-frame context.

Compared to DualPM (Kaye et al., 2025), which uses single frames, CEFFYL achieves consistent improvements of $+7.9$, $+7.2$, and $+8.5$ points across the three datasets. This confirms the advantage of our architecture, particularly the temporal layers. We further confirm this by re-training DualPM on the same data as CEFFYL in Section 4.4.

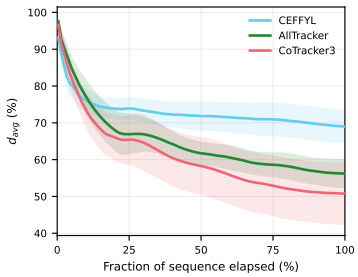


Figure 5: Tracking accuracy d_{avg} vs. fraction of sequence elapsed.

To analyse long-term robustness, Figure 5 plots tracking performance (d_{avg}) over time for the five longest Horse10 sequences. CEFFYL maintains performance for significantly longer, finishing 13 and 18 points higher than AllTracker and CoTracker3.

Table 2 reports occlusion accuracy (OA) and average Jaccard (AJ). CEFFYL outperforms AllTracker by $+10.0$ and $+11.7$ points in OA on Horse10 and PFERD, proving the efficacy of our amodal predictions at understanding occlusions. On Showground, where SAM3 actually harms AllTracker's visibility metrics, CEFFYL outperforms the baseline by $+23.7$ in OA and $+21.4$ in AJ.

Zero-Shot Category Generalisation. CEFFYL is trained specifically on horses. However, we show strong generalisation for dense correspondence on a number of other animals in Figure 6. The points are simply coloured by their position in the canonical space. The pointmap reconstructions also highlight how the model warps the shape to fit new instances (e.g. notice the humps on the camel, the protruding of the tail of the dog, and the flattening of the face for the koala). We also draw attention to failure cases: the elephant shows good correspondence but the trunk fails to materialise in the pointmap, and although for the flamingo the model warps the point map to the body and head with surprising success, it struggles to track the thin legs.

4.4 Ablations

We perform our ablations using Horse10 and Showground, covering both short and long sequences.

Training on VarenPoser Data. To understand the effect of our training data, we use the same VarenPoser Lyu et al. (2026) data to train both AllTracker (Harley et al., 2025) (our best point tracking competitor) and DualPM (Kaye et al., 2025). See details of this training in the supplementary

Table 3: Ablation results for (a) the effect of VarenPoser training data, (b) the contribution of individual model components on d_{avg}, (c) descriptor training and (d) employing global attention in the temporal layers or different numbers of learnt aggregation tokens. SG denotes the Showground.

(a) Effect of VarenPoser data

Model	Train Data	Horse10	SG
AllTracker	Kubric+mix	54.3	53.5
AllTracker	VarenPoser F.T.	<u>58.4</u>	41.9
DualPM	Animodel	55.3	<u>61.7</u>
DualPM	VarenPoser	52.3	55.6
CEFFYL	VarenPoser	**63.2**	**70.2**

(b) Component ablation

Spat. A.	Temp. A.	Desc. Head	Horse10	SG
✓			55.4	62.7
✓	✓		61.9	69.4
✓	✓	✓	**63.2**	**70.2**

(c) Descriptor training

Training	Inference	Horse10	SG
Joint	NN	60.3	68.3
Joint	NN + Desc.	61.8	69.0
Two-Stages	NN	<u>61.9</u>	<u>69.4</u>
Two-Stages	NN + Desc.	**63.2**	**70.2**

(d) Global attention vs. learnt tokens

Model	No. Tokens	Horse10	SG
Global Attn	0	59.5	67.8
Learnt Agg. Tokens	16	61.5	**70.3**
Learnt Agg. Tokens	32	<u>61.7</u>	68.8
Learnt Agg. Tokens	64	**61.9**	<u>69.4</u>
Learnt Agg. Tokens	128	60.4	69.2

Section A.4. Table 3a shows that fine-tuning AllTracker increases performance on Horse10 but still underperforms CEFFYL by 4.8 points, while it harms performance on Showground by 11.6 points. This confirms that object-specific training data is not the sole reason for our boost in performance. When re-training DualPM with VarenPoser, performance drops.

Component Contributions. We determine the contribution of each of our major architecture choices in Table 3b. Adding temporal attention, we see consistent performance increases of $+6.5$ points for Horse10 and $+6.7$ points for Showground benchmarks. Adding the descriptor head, we see a boost of $+1.3$ points on Horse10 and $+0.8$ point increase on Showground, likely due to the subjects filling a smaller portion of the screen which means the top-k can include more distractors.

Descriptor Training Strategies. We ablate our two-stage training in Table 3c. For joint training, we add the infoNCE loss $\mathcal{L}_{desc}$ to the main loss $\mathcal{L}_{stage1}$. The descriptor head consistently adds performance over the models without it. Best performance is achieved with our two-stage training.

Learnt Aggregation vs. Global Attention. We ablate the contribution of our temporal attention module over global attention. Our global attention model is similar to VGGT (Wang et al., 2025a): global attention layers are interleaved between frame attention layers, and the outputs are concatenated before entering the DPT head. From Table 3d, our temporal aggregation attention outperforms global attention layers by $+2.4$ points. We also ablate the number of learned tokens: 64 learned tokens marginally outperforms 16 and 32, and 128 is worse.

Additional ablations on model capacity, descriptor top-*k*, descriptor dimension, DINOv2 backbone size, training-pair strides and detection jitter are in the supplementary Section A.5.

Limitations Although CEFFYL shows impressive zero-shot dense correspondence for even for unseen animals, it requires separate models or a more diverse training dataset to extend to other categories. This dense matching is also more expensive and time-consuming to compute when compared to smaller, generalisable point trackers (see supplementary Section A.2). Lastly, although visibility can be predicted with amodal layer predictions, this can falter when the object is heavily occluded, and it cannot reliably handle occlusions by other objects.

5 Conclusion

CEFFYL maps pairs of images to amodal pointmap representations in a shared canonical space. This allows the model to understand the shape and pose of the object as well as how the views relate to one another. Our work is able to achieve state-of-the-art performance on three category-specific benchmarks, outperforming generalisable point trackers, correspondence models, 4D reconstruction methods, and also prior work which attempted canonical mapping using a single frame. Our work also introduces a new challenging benchmark called Showground, which evaluates long-term point tracking on keypoints of running and jumping horses. We show robust zero-shot dense correspondence generalisation to other quadruped and biped animals and present extensive ablations to support the efficacy of our architecture.

AI Use Statement. We used generative AI tools to refine equation annotations and provide additional relevant prior works. We have reviewed all of the AI assisted work. We do not use generative AI tools for datasets, helping to develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypothesis, design or provide feedback on research methodology or experiments, implement methods, assist with translation, clean and reformat datasets, support qualitative and thematic data analysis or interpret results. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M Seitz, and Richard Szeliski. Building rome in a day. *Communications of the ACM*, 2011.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris Coll-Vinent, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. Sam 3: Segment anything with concepts. In *International Conference on Learning Representations (ICLR)*, 2026.

Ho Kei Cheng and Alexander G Schwing. Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model. In *European Conference on Computer Vision (ECCV)*, 2022.

Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang. Rethinking space-time networks with improved memory coverage for efficient video object segmentation. *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Seokju Cho, Sunghwan Hong, and Seungryong Kim. Cats++: Boosting cost aggregation with convolutions and transformers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

Claudia Cuttano, Gabriele Trivigno, Carlo Masone, and Stefan Roth. MARCO: Navigating the unseen space of semantic correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Ahmad Darkhalil, Rhodri Guerrier, Adam W Harley, and Dima Damen. Egopoints: Advancing point tracking for egocentric videos. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2025.

Carl Doersch, Ankush Gupta, Larisa Markeeva, Adria Recasens, Lucas Smaira, Yusuf Aytar, Joao Carreira, Andrew Zisserman, and Yi Yang. Tap-vid: A benchmark for tracking any point in a video. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Carl Doersch, Pauline Luc, Yi Yang, Dilara Gokay, Skanda Koppula, Ankush Gupta, Joseph Heyward, Ignacio Rocco, Ross Goroshin, Joao Carreira, et al. Bootstap: Bootstrapped training for tracking-any-point. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, 2024.

Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *European Conference on Computer Vision (ECCV)*, 2014.

Haiwen Feng, Junyi Zhan*, Qianqian Wang, Yufei Ye, Pengcheng Yu, Michael J. Black, Trevor Darrell, and Angjoo Kanazawa. St4rtrack: Simultaneous 4d reconstruction and tracking in the world. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 2013.

Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, et al. Kubric: A scalable dataset generator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Antoine Guédon, Shu Nakamura, Nicolas Dufour, Jiahui Lei, Ko Nishino, and Angjoo Kanazawa. Surflo: Consistent 3d surface flow model with global state. *arXiv preprint arXiv:2606.13644*, 2026.

Rhodri Guerrier, Adam W Harley, and Dima Damen. Pointst3r: Point tracking through 3d grounded correspondence. In *2026 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2026.

Adam W Harley, Zhaoyuan Fang, and Katerina Fragkiadaki. Particle video revisited: Tracking through occlusions using point trajectories. In *European Conference on Computer Vision (ECCV)*, 2022.

Adam W Harley, Yang You, Xinglong Sun, Yang Zheng, Nikhil Raghuraman, Yunqi Gu, Sheldon Liang, Wen-Hsuan Chu, Achal Dave, Suya You, et al. Alltracker: Efficient dense point tracking at high resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 1981.

Zhaoyang Huang, Xiaoyu Shi, Chao Zhang, Qiang Wang, Ka Chun Cheung, Hongwei Qin, Jifeng Dai, and Hongsheng Li. Flowformer: A transformer architecture for optical flow. In *European Conference on Computer Vision (ECCV)*, 2022.

Tomas Jakab, Ruining Li, Shangzhe Wu, Christian Rupprecht, and Andrea Vedaldi. Farm3d: Learning articulated 3d animals by distilling 2d diffusion. In *Proceedings of the International Conference on 3D Vision (3DV)*, 2024.

Yanqin Jiang, Tengfei Wang, Zhengwei Wang, Chenjie Cao, Junta Wu, Wenhan Luo, Weiming Hu, Jin Gao, and Chunchao Guo. Omnix: Any-view and any-time 4d reconstruction via feed-forward trajectory fields. In *European Conference on Computer Vision (ECCV)*, 2026.

Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker: It is better to track together. In *European Conference on Computer Vision (ECCV)*, 2024.

Nikita Karaev, Yuri Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker3: Simpler and better point tracking by pseudo-labelling real videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

Ben Kaye, Tomas Jakab, Shangzhe Wu, Christian Ruprecht, and Andrea Vedaldi. Dualpm: Dual posed-canonical point maps for 3d shape and pose reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Nilesh Kulkarni, Abhinav Gupta, David F Fouhey, and Shubham Tulsiani. Articulation-aware canonical surface mapping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *European Conference on Computer Vision (ECCV)*, 2024.

Ci Li, Ylva Mellbin, Johanna Krogager, Senya Polikovsky, Martin Holmberg, Nima Ghorbani, Michael J Black, Hedvig Kjellström, Silvia Zuffi, and Elin Hernlund. The poses for equine research dataset (pferd). *Scientific Data*, 2024a.

Zizhang Li, Dor Litvak, Ruining Li, Yunzhi Zhang, Tomas Jakab, Christian Rupprecht, Shangzhe Wu, Andrea Vedaldi, and Jiajun Wu. Learning the 3d fauna of the web. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024b.

Yanbin Liu, Linchao Zhu, Makoto Yamada, and Yi Yang. Semantic correspondence as an optimal transport problem. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Bruce D Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI'81: 7th International Joint Conference on Artificial intelligence*, 1981.

Yihang Luo, Shangchen Zhou, Yushi Lan, Xingang Pan, and Chen Change Loy. 4rc: 4d reconstruction via conditional querying anytime and anywhere. In *International Conference on Machine Learning (ICML)*, 2026.

Jin Lyu, Liang An, Pujin Cheng, Yebin Liu, and Xiaoying Tang. 4dequine: Disentangling motion and appearance for 4d equine reconstruction from monocular video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Alexander Mathis, Tanja Biasi, Steffen Schneider, and Mackenzie W. Mathis. Pretraining boosts out-of-domain robustness for pose estimation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2021.

Lukas Mehl, Jenny Schmalfuss, Azin Jahedi, Yaroslava Nalivayko, and Andrés Bruhn. Spring: A high-resolution high-detail dataset and benchmark for scene flow, optical flow and stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Juhong Min, Jongmin Lee, Jean Ponce, and Minsu Cho. Hyperpixel flow: Semantic correspondence with multi-layer neural features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: A versatile and accurate monocular slam system. *IEEE Rransactions on Robotics*, 2015.

Tuan Ngo, Peiye Zhuang, Evangelos Kalogerakis, Chuang Gan, Sergey Tulyakov, Hsin-Ying Lee, and Chaoyang Wang. Delta: Dense efficient long-range 3d tracking for any video. In *International Conference on Learning Representations (ICLR)*, 2025.

Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research Journal*, 2024.

René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. In *International Conference on Learning Representations (ICLR)*, 2025.

Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Xiaoyu Shi, Zhaoyang Huang, Dasong Li, Manyuan Zhang, Ka Chun Cheung, Simon See, Hongwei Qin, Jifeng Dai, and Hongsheng Li. Flowformer++: Masked cost volume autoencoding for pretraining optical flow estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 2024.

Edgar Sucar, Zihang Lai, Eldar Insafutdinov, and Andrea Vedaldi. Dynamic point maps: A versatile representation for dynamic 3d reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

Edgar Sucar, Eldar Insafutdinov, Zihang Lai, and Andrea Vedaldi. V-dpm: 4d video reconstruction with dynamic point maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Xinglong Sun, Adam W Harley, and Leonidas J Guibas. Refining pre-trained motion models. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision (ECCV)*, 2020.

Narek Tumanyan, Assaf Singer, Shai Bagon, and Tali Dekel. Dino-tracker: Taming dino for self-supervised point tracking in a single video. In *European Conference on Computer Vision (ECCV)*, 2024.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *ArXiv*, abs/1807.03748, 2018.

Mel Vecerik, Carl Doersch, Yi Yang, Todor Davchev, Yusuf Aytar, Guangyao Zhou, Raia Hadsell, Lourdes Agapito, and Jon Scholz. Robotap: Tracking arbitrary points for few-shot visual imitation. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2025a.

Jianyuan Wang, Minghao Chen, Shangzhan Zhang, Nikita Karaev, Johannes Schönberger, Patrick Labatut, Piotr Bojanowski, David Novotny, Andrea Vedaldi, and Christian Rupprecht. Vggt-. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025b.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024a.

Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. Tartanair: A dataset to push the limits of visual slam. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

Yihan Wang, Lahav Lipson, and Jia Deng. Sea-raft: Simple, efficient, accurate raft for optical flow. In *European Conference on Computer Vision (ECCV)*, 2024b.

Shangzhe Wu, Ruining Li, Tomas Jakab, Christian Rupprecht, and Andrea Vedaldi. Magicpony: Learning articulated 3d animals in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Tianhao Wu, Chuanxia Zheng, Frank Guan, Andrea Vedaldi, and Tat-Jen Cham. Amodal3r: Amodal 3d reconstruction from occluded 2d images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

Yuxi Xiao, Qianqian Wang, Shangzhan Zhang, Nan Xue, Sida Peng, Yujun Shen, and Xiaowei Zhou. Spatialtracker: Tracking any 2d pixels in 3d space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Zongxin Yang, Yunchao Wei, and Yi Yang. Associating objects with transformers for video object segmentation. *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Dingxi Zhang, Fangjinhua Wang, Marc Pollefeys, and Haofei Xu. Megaflow: Zero-shot large displacement optical flow. In *European Conference on Computer Vision (ECCV)*, 2026.

Junyi Zhang, Charles Herrmann, Junhwa Hur, Luisa Polania Cabrera, Varun Jampani, Deqing Sun, and Ming-Hsuan Yang. A tale of two features: Stable diffusion complements dino for zero-shot semantic correspondence. *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Junyi Zhang, Charles Herrmann, Junhwa Hur, Eric Chen, Varun Jampani, Deqing Sun, and Ming-Hsuan Yang. Telling left from right: Identifying geometry-aware semantic correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Yang Zheng, Adam W Harley, Bokui Shen, Gordon Wetzstein, and Leonidas J Guibas. Pointodyssey: A large-scale synthetic dataset for long-term point tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

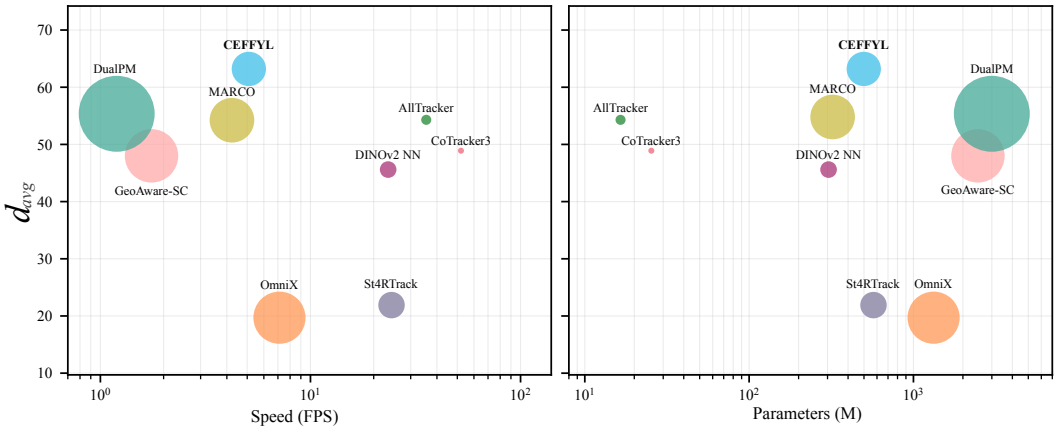


Figure 7: Tracking accuracy d_{avg} on Horse10 against inference speed (left) and the model parameters (right). Bubble area is proportional to GFLOPs per frame.

A Appendix

A.1 Supplementary Video

As part of the supplementary, we provide a companion video showcasing a number of qualitative examples of our model, CEFFYL. The video begins with visualisations of the visible layer predictions and then the amodal layers are added in different colours. We then show the query and target reconstructions with layer colouring over a sequence of frames with interesting horse motion, taken from Showground. Next, the video showcases dense correspondence. Points in the image and pointmaps are simply coloured based on their position in the shared canonical space. We showcase examples of this for four sequences taken from Showground and for other, unseen animals at the end of the video. We also showcase qualitative examples of point tracking for Horse10 and PFERD videos. Red lines in these sections denote error from the ground truth and we compare against the best point tracker competitor, AllTracker (Harley et al., 2025).

A.2 Model Speed & Size

Figure 7 reports comparisons on speed and number of parameters for CEFFYL and a number of correspondence, 4D reconstruction and point tracking models. Correspondence models MARCO (Cuttano et al., 2026) and GeoAware-SC (Zhang et al., 2024) both underperform our work, CEFFYL, on tracking accuracy whilst also being slower during inference. MARCO is only slightly smaller on parameters, whilst GeoAware-SC is a much larger model. Although 4D reconstruction models OmniX (Jiang et al., 2026) and St4RTrack (Feng et al., 2025) are slightly faster during inference and roughly comparable on parameter size, they come no where near the tracking accuracy achieved by CEFFYL. Point trackers, such as AllTracker (Harley et al., 2025) and CoTracker3 (Karaev et al., 2025), do not perform 3D object modelling, so they are therefore more lightweight and faster than our work. However, they trade tracking accuracy for this advantage. The work of DualPM (Kaye et al., 2025), which is the closest work to ours, is slower, larger and less accurate at tracking than CEFFYL.

A.3 Addressing PFERD-VarenPoser Motion Overlap

We use the VarenPoser dataset (Lyu et al., 2026) to generate our amodal pointmaps and to train CEFFYL. However, the motion of the synthetic subjects in the VarenPoser dataset is based on the dynamics of the PFERD (Li et al., 2024a) motion capture markers. As we use sequences of PFERD in our main evaluation, it is important to disentangle the effect of this training similarity on the evaluation. Specifically, we construct a second PFERD evaluation with 10 240-frame sequences taken from the ID=3 miniature horse, as this is the only unseeen subject by the VarenPoser dataset. The results of this additional evaluation can be seen in Table 4. As expected, AllTracker (Harley

Table 4: Tracking accuracy d_{avg} on 10 240-frame sequences from ID=3 sequences in the PFERD dataset. This subject has specifically never been seen by the VarenPoser dataset, used in our training.

Model	d_{avg}
AllTracker (Harley et al., 2025)	55.5
AllTracker + SAM3	67.3
DualPM (Kaye et al., 2025)	<u>81.1</u>
CEFFYL	**88.6**

et al., 2025) performs roughly similar as on the original PFERD evaluation (which uses three horse subjects, two of which seen by VarenPoser), and a similar message exists for the efficacy of using SAM3 for increasing tracking performance. DualPM (Kaye et al., 2025) outperforms AllTracker by a large margin of 13.8 percentage points, as expected and consistent with the 11.3 point increase we see in the main evaluation. Our model, CEFFYL, increases this gain further with increases of 7.5 and 21.3 percentage points over DualPM and AllTracker, respectively. This additional evaluation shows that the boost in performance over point trackers and object-centric models, is not due to the VarenPoser and PFERD overlap, as this evaluation specifically targets a PFERD horse subject unseen by the training data.

A.4 VarenPoser Ablation Training Details

In the ablations Section 4.4, we discuss the effect that VarenPoser training data has on AllTracker (Harley et al., 2025) and DualPM Kaye et al. (2025). Specifically, we fine-tune AllTracker for 5k steps (performance does not improve with longer training) using 5k 24-frame VarenPoser clips (where the posed mesh is used to extract point tracks as in Zheng et al. (2023)) mixed with 5k samples from Kubric (Greff et al., 2022). For DualPM, we follow the original implementation details and train end-to-end with 30k samples for 100k training steps. We train DualPM from scratch as the canonical shapes of the Animodel (Jakab et al., 2024) and VarenPoser datasets do not match.

A.5 Additional Ablations

Table 5: Ablation results for model capacity, top-k re-ranking, and descriptor dimension. SG denotes the Showground dataset.

(a) Model capacity

Spat. Layers	Temp. Layers	Horse10	SG
1	1	60.1	68.0
2	2	61.3	66.8
4	4	<u>61.9</u>	**69.4**
8	4	**62.0**	68.0
16	4	61.0	<u>68.9</u>

(b) Top-k re-ranking

Top-k	Horse10	SG
1	61.9	69.4
5	62.7	<u>70.0</u>
10	<u>63.0</u>	<u>70.0</u>
25	**63.2**	**70.2**

(c) Descriptor dimension

Dim.	Horse10	SG
8	62.6	**70.2**
16	<u>62.9</u>	69.3
32	**63.2**	<u>69.9</u>
64	**63.2**	**70.2**
128	63.1	69.8

Model Capacity. We also ablate the effect that the number of spatial and temporal attention layers has on the tracking performance. Best performance is achieved with 4 to 8 spatial attention layers and 4 temporal attention layers.

Descriptor Top-K. We ablate the effect of top-k value in our descriptor re-ranking stage of inference in Table 5b. All values of $k > 1$ increase performance over the model without the descriptors ($k = 1$). Increasing $k > 5$ pushes performance slightly. This is consistent across both datasets.

Descriptor Dimension. The effect of descriptor dimension length on tracking accuracy is reported in Table 5c. Increasing the number of tokens does seem to have an effect on tracking performance. However, this clearly peaks around 64 and begins to fall off when moving up to 128. Descriptor dimension $F = 64$ is the best option.

Frame Strides. During training, we present the model with two frames, a query and a target. As our frames are of a temporally consistent subject, we choose to ablate the effect that stride length has on tracking accuracy. We sample frame pairs from each stride value as in MASt3R Leroy et al. (2024), whereby the larger stride values are weighted higher for selection. The results in Table 6

Table 6: Ablation results for the DINOv2 backbone size and training-pair stride length.

(a) DINOv2 backbone size

Backbone	Params.	Horse10	Showground
ViT-B	86M	58.4	64.3
ViT-L	304M	**61.9**	69.4
ViT-G	1.1B	61.1	**71.0**

(b) Training-pair stride length

Strides	Horse10	Showground
[5]	60.4	68.3
[5, 10, 15]	61.6	67.7
[5, 10, 15, 20, 25]	**61.9**	**69.4**
[5, 10, 15, 20, 25, 30, 35]	61.5	69.3
[5, 10, 15, 20, 25, 30, 35, 40, 45]	61.6	68.1

show that as stride length increases, so does performance. However, the gains seem to plateau after a maximum gap of 25 frames between the query and the target.

DINOv2 Backbone. We ablate the effect that the frozen DINOv2 feature extractor has on the tracking performance of our model. Specifically, we ablate over the base, large and giant variations of the DINOv2 model. From 6, both the large and giant variants offer a clear increase over the base model. ViT-L is chosen as although is performs worse on Showground, it performs better on Horse10 and reduces parameters considerably.

Table 7: Random detection jitter results for AllTracker and CEFFYL when using cropped image input.

Model	(0,25)	(0.50)	(0,100)
AllTracker (Harley et al., 2025)	52.3	50.6	49.1
CEFFYL	60.3	58.7	55.9

Detection Jitter. Due to the reliance that our model has on mask detections for cropping around the object, we ablate the effect that the detection stability has on downstream tracking performance. We conduct this ablation in two ways. Firstly, we record the tracking accuracy of both AllTracker (Harley et al., 2025) (with the SAM3 detector) and our work for Horse10 when changing the number of pixels between the object and the edge of the crop. It is clear from Figure 8 that performance drops as the buffer increases and the crop is wider around the object. However, we see that this effect also occurs for AllTracker. In fact, the gap between our model and AllTracker stays relatively stable(e.g. $+8.9$ for a buffer of 10 pixels and $+7.9$ for a buffer of 50 pixels). Secondly, we evaluate the tracking performance when we add a random jitter to the crop. Specifically, Table 7 shows results for when the crop buffer (as explained previously) has a random value between 0 and $[25, 50, 100]$. As can be seen, results again drop as the jitter becomes more unstable, with a drop of 4.4 between the minimum and maximum of the random range. We also see a similar drop between the minimum and maximum for AllTracker of 3.2 percentage points. Both of these results show that despite the models being particularly sensitive to detection noise, our model still holds its competitive edge over the state-of-the-art AllTracker.

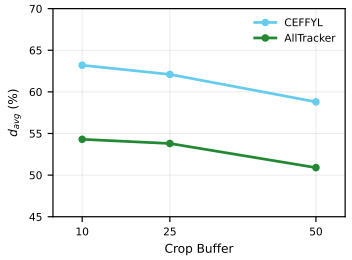


Figure 8: Tracking accuracy d_{avg} on Horse10 when increasing the crop buffer.

A.6 Showground Annotations

We present examples of Showground annotations for a sample of evaluation frames in Figure 9. The benchmark includes 9 videos ranging from 701 to 1961 frames where the first frame is our query and then 10 evaluation frames are annotated across the length of the video. In total we have 90 evaluation frames, 610 evaluation points of which 497 are marked as visible.



Figure 9: Example of annotations on Showground. Each row is an example with a few frames for reference. Coloured points represent the ground truth and unfilled points are occluded.