MuM: Multi-View Masked Image Modeling for 3D Vision

David Nordström[1] Johan Edstedt[2] Fredrik Kahl[1] Georg Bökman[3]

[1]Chalmers University of Technology [2]Linköping University [3]University of Amsterdam

Abstract

Self-supervised learning on images seeks to extract meaningful visual representations from unlabeled data. When scaled to large datasets, this paradigm has achieved state-of-the-art performance and the resulting trained models such as DINOv3 have seen widespread adoption. However, most prior efforts are optimized for semantic understanding rather than geometric reasoning. One important exception is Cross-View Completion, CroCo, which is a form of masked autoencoding (MAE) tailored for 3D understanding. In this work, we continue on the path proposed by CroCo and focus on learning features tailored for 3D vision. In a nutshell, we extend MAE to arbitrarily many views of the same scene. By uniformly masking all views and employing a lightweight decoder with inter-frame attention, our approach is inherently simpler and more scalable than CroCo. We evaluate the resulting model, MuM, extensively on downstream tasks including feedforward reconstruction, dense image matching and relative pose estimation, finding that it outperforms the state-of-the-art visual encoders DINOv3 and CroCo v2. Code is available at https://github.com/davnords/mum.

1. Introduction

Self-supervised learning (SSL) enables large-scale pretraining without manual labels and has become a cornerstone of modern visual representation learning. A popular family of SSL objectives for image data are based on masked autoencoding (MAE) [29]. In MAE, the training task consists of reconstructing a partially masked image.

Pixel reconstruction is an inherently low-level dense objective. To increase 3D understanding, CroCo [66] proposed conditioning the reconstruction with an unmasked reference view from the same scene. However, this objective relies on substantial overlap between pairs, making the data sampling fragile, as pointed out by Loiseau et al. [37]. To alleviate the sampling issue, Loiseau et al. [37] reformulated the task as a co-visibility segmentation objective. On the other hand, relying on co-visibility necessitates access to ground-truth geometry, limiting its applicability for SSL.

Other approaches for SSL on images include the state-of-the-art DINO family [12, 40], with DINOv3 [50] as its latest incarnation. DINOv3 produces rich image representations and is trained through self-distillation on billions of unlabeled images.

Recent works in 3D vision have successfully used DINOv2/3 networks as feature encoders [62], but the learned features in DINO are commonly described as semantic rather than geometric. Furthermore, DINOv3 relies on carefully crafted heuristics such as Sinkhorn-Knopp centering to avoid training collapse and the requirement of a very large dataset makes it an inaccessible method for most of the academic community.

Motivated by the challenges outlined above, this work proposes a simple SSL objective for learning geometric features for 3D tasks while enabling flexible data sampling. We show that extending the MAE objective to arbitrarily many views of the same scene yields a simple and powerful supervision strategy, surpassing DINOv3 in 3D vision pipelines while requiring roughly $30\times$ less training compute.[1]

We call our objective, as well as the trained network, MuM (**Mu**lti-View **M**asked Image Modeling). MuM achieves strong downstream performance on a variety of tasks and is illustrated in Figure 1. The main contributions are as follows:

1. We extend masked image modeling to an arbitrary number of views of a scene and propose a simple, scalable training framework tailored for 3D vision.
2. We perform a comprehensive comparison of a wide range of models through dense feature matching performance using a linear probe.
3. We validate our approach on 3D vision tasks such as feedforward reconstruction, feature matching, and pose estimation by comparing MuM as a feature extractor to the state-of-the-art baselines DINOv3 and CroCo v2. In particular, we train multiple feedforward reconstruction models inspired by, albeit smaller scale than, VGGT [62]. Further, we train state-of-the-art two-view matchers following RoMa [23].

[1]61,440 H100 hours (DINOv3-7B) vs. 4,608 A100 hours (MuM).

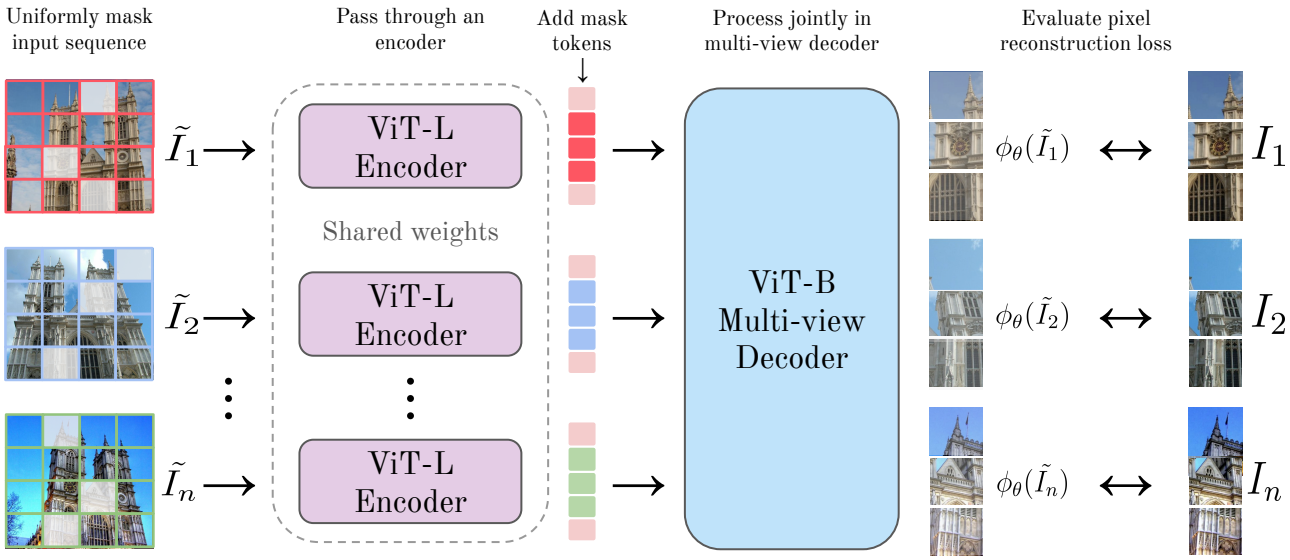


Figure 1. **MuM SSL pretraining**. An arbitrary number of input images from the same scene are first uniformly masked and processed independently in a ViT-L encoder. The representations are then jointly processed in a multi-view ViT-B decoder. The final representations are linearly mapped to pixel space and the reconstruction loss is computed according to Equation 1.

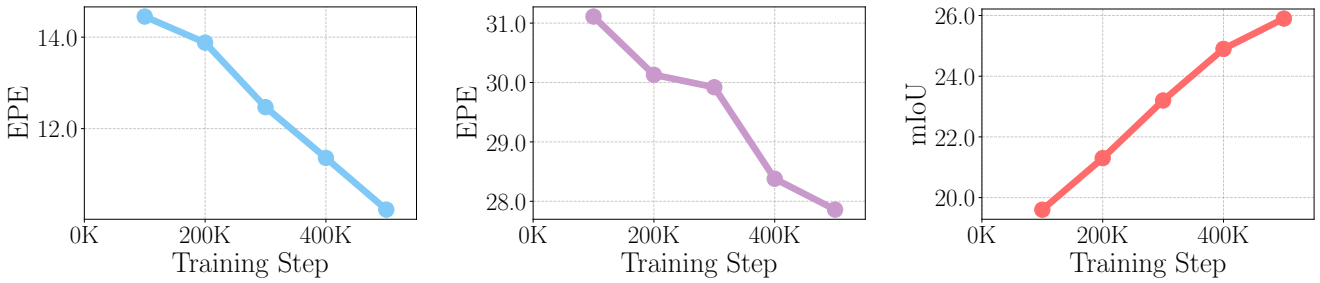
2. Related work

Masked Image Modeling (MIM). The success of masked language modeling for self-supervised pre-training in NLP [18, 41, 57] inspired similar efforts in computer vision. Following the popularization of ViTs [20], patch-based masking techniques were revisited to great success [4, 25, 29, 70]. He et al. [29] introduced the masked auto-encoder (MAE) which uses a pixel reconstruction objective as its pre-text task. The MAE reconstruction objective has been extended to, among other things, videos [13, 56, 63] and multi-modal data [4]. MAE for videos has further been generalized to videos from multiple views [48, 49], with similar motivations as our work. Instead of videos, we focus on image data and, in particular, downstream applications involving unstructured and sparse sequences of images from a scene.

Closest to our work, Weinzaepfel et al. [66] extended the MAE objective through cross-view completion (CroCo), and subsequently CroCo v2 [67], by conditioning on an unmasked reference view to encourage geometric understanding. In this work, we also aim to learn features that are useful for 3D downstream tasks. In contrast to CroCo, we sample an arbitrary number of frames with larger viewpoint variations and use alternating attention across them, yielding a simple, symmetric architecture that supports both single- and multi-frame training while offering greater robustness to frame sampling.

A different popular approach to SSL relies on aligning a student's features with those of a teacher. The most popular such approach relies on training the student and teacher from scratch jointly, often having the teacher be an exponential-moving-average of the student [55]. Similar to MAE, one can use a masked reconstruction objective in this framework, but the reconstruction being in feature space rather than pixels [16, 40, 80]. Combing this approach with instance discrimination [12] has lead to the current state-of-the-art SSL objective, DINOv3 [50]. Although effective at producing highly semantic features, it is computationally demanding, as it relies on a student-teacher setup and attends between masked patches. In this work, we study how this objective transfers to geometric downstream tasks and, interestingly, conclude that a simple pixel reconstruction objective works better at learning geometric features. We demonstrate the effectiveness of our objective on a range of 3D computer vision applications.

Multiple-view Geometry. In an effort to replace hand-crafted methods for multiple-view geometry [28] with learning-based approaches, two-view neural network architectures have been used in image matching [22, 23, 35, 52], pose estimation [19] and stereo reconstruction [10, 64]. To extend its usefulness in traditional 3D computer vision tasks such as structure-from-motion (SfM), architectures processing arbitrarily many frames jointly were introduced [8, 11, 33, 54, 61, 62, 71, 74]. Recently, transformer architectures trained on large-scale 3D-annotated datasets have been proposed to jointly predict all scene attributes in a single forward pass, notable examples being VGGT [62] and MapAnything [33]. VGGT allows inter-frame communication through alternating frame-wise and global atten-



(a) MegaDepth linear ↓ (b) ScanNet linear ↓ (c) ADE20K k-NN ↑

Figure 2. **Training dynamics.** Dense matching with a linear probe and semantic segmentation performance during 500K steps of training.

tion. In this work, we adopt the same attention mechanism between frames in our decoder but in the context of SSL. We further propose a symmetric architecture, rather than anchoring a reference view, similar to Wang et al. [65].

3. Method

We begin by formalizing multi-view masked image modeling (Section 3.1), followed by introducing our proposed learning objective (Section 3.2), architecture (Section 3.3), and training procedure (Section 3.4).

3.1. Problem definition

Standard masked image modeling considers reconstructing an image from a partially masked version of itself. We adopt a more general formulation that naturally extends this idea to multiple views.

Let $\mathcal{I} = (I_1, I_2, \ldots, I_n)$ be a sequence of n images of the same scene, each divided into N non-overlapping patches. For each view i, we mask a subset of the patches according to binary masks $M_i \in \{0,1\}^N$, with masking ratios $\gamma_i \in [0,1]$. We interpret M_i as being 1 for the masked patches and obtain a partially masked sequence of images

$$\tilde{I}_i = \overline{M}_i \odot I_i, \quad \overline{M}_i = 1 - M_i,$$

where $\odot$ denotes element-wise patch selection.

The learning objective is to train a neural network ϕ_θ to predict a target representation $f(\mathcal{I})$ from the visible patches in $\tilde{\mathcal{I}} = (\tilde{I}_1, \tilde{I}_2, \ldots, \tilde{I}_n)$. The loss (up to a normalizing constant) is given by:

$$\mathcal{L}(\theta) = \sum_{i=1}^{n} ||M_i \odot (\phi_\theta(\tilde{I}_i) - f(I_i))||^2. \qquad (1)$$

The role of f is to specify the reconstruction target. In the simplest case, f is the identity (possibly with normalization), yielding pixel-space regression as in MAE or CroCo. More generally, f may produce higher level representations, which shows the close connection to self-distilled teacher neworks, as in iBOT, DINOv2, and CAPI [16], as well as frozen pretrained teachers, as in BeiT [5] and EVA [26].

The original MAE objective corresponds to the single-view setting ($n = 1$) and a fixed masking ratio $\gamma_1 = 0.75$ whereas the cross-view completion formulation in CroCo uses two views ($n = 2$) with $\gamma_1 = 0.9$ and $\gamma_2 = 0$.

3.2. Learning objective

We design MuM by uniformly sampling sequences between $n = 2$ and $n = 24$ frames and using a uniform masking ratio of $\gamma_i = 0.75$ for all i. The target representation f simply normalizes the pixel values in a patch by its mean and standard deviation. Our objective is therefore exactly equal to MAE for single views ($n = 1$). In this way, we address the difficulties in CroCo of sampling co-visible pairs as there is always a fallback to the standard MAE objective. Furthermore, our objective straightforwardly generalizes to an arbitrary number of views, while it is not obvious how to generalize CroCo to more than two views in terms of selecting the masking ratios γ_i. We ablate our design choices and the objective in Section 4.7.

3.3. Network architecture

We implement ϕ_θ as a Vision Transformer (ViT) [20] with an encoder-decoder structure (cf. Figure 1), similar to MAE and CroCo. For the ViT, we use modern architectural choices such as axial RoPE [50, 51]. Each input image $I_i \in \mathcal{I}$ is patchified into N tokens. Following Section 3.1, a fraction γ_i of the patches is removed, and the remaining tokens are passed through a ViT encoder. Thereafter, learnable mask tokens are appended for missing patches. These tokens are then processed jointly in a ViT-B decoder using $L = 6$ alternating attention blocks. Each block performs (i) frame-wise attention, restricting attention within each view, followed by (ii) global attention, allowing tokens to attend across all views. This symmetric design avoids designating any reference frame. Lastly, the token embeddings for each image are passed through a linear prediction head that regresses the normalized RGB values for each patch and the loss is computed by Equation 1.

3.4. Training

Implementation Details. We train MuM for 500k steps by minimizing the training loss (1) using the AdamW [38] optimizer. We use a cosine learning rate scheduler and a linear warmup of 25K steps. Following the linear scaling rule of Goyal et al. [27], $\text{lr} = \frac{\text{blr}}{256} \times \text{batch size}$, we use a base learning rate of 1×10^{-4}, yielding a peak rate of 2.4×10^{-3} for our global batch size of 6144. Similar to VGGT, for each batch we randomly select a sequence length uniformly between 2 and 24. Based on this length, we then sample as many training scenes as possible without exceeding 96 total frames per GPU. We resize the images to 256×256 and randomly apply horizontal mirroring augmentation per frame. Pre-training on $64 \times$A100 GPUs takes roughly three days. As shown in Figure 2, the evaluation metrics consistently improve during training.

Training Data. We train on a diverse set of 3D datasets inspired by the mix used in DUSt3R [64] and VGGT. In particular, we train on 3DStreetView [75], ARKitScenes [6], BlendedMVS [72], CO3D [43], DL3DV10K [36], Hypersim [44], MegaDepth [34], MegaScenes [59], ScanNet++ [73], RealEstate10K [81], and WildRGBD [69]. When applicable, we utilize only the training splits. In total, the collection comprises around 20 million individual frames (see Table 1 for a detailed breakdown). In Figure 3, we visualize a representative sequence sampled from our data collection. MuM provides a simple symmetric objective which enables us to mix multi-view and single-view data. Inspired by Siméoni et al. [50], we use a 10% probability to sample data from ImageNet-1K [17, 45]. Another advantage of our architecture is that it is robust to simpler sampling schemes and does not strictly require covisibility.

Table 1. **Dataset mixture** and sample sizes for MuM pre-training.

Datasets	Type	Prob.	Frames
3DStreetView [75]	Outdoor / Real	10%	6282k
WildRGBD [69]	Objects / Real	10%	3352k
DL3DV10K [36]	Outdoor / Real	10%	3317k
RealEstate10K [81]	Indoor / Real	10%	1332k
ImageNet1K[2] [17, 45]	Objects / Real	10%	1281k
MegaScenes [59]	Outdoor / Real	10%	1235k
ScanNet++ [73]	Indoor / Real	10%	1045k
CO3D [43]	Objects / Real	5%	546k
MegaDepth [34]	Outdoor / Real	10%	205k
BlendedMVS [72]	Aerial / Synthetic	5%	115k
ARKitScenes [6]	Indoor / Real	5%	113k
Hypersim [44]	Indoor / Synthetic	5%	43k
Total			**19462k**

[2]ImageNet1K consists of only single-view data.

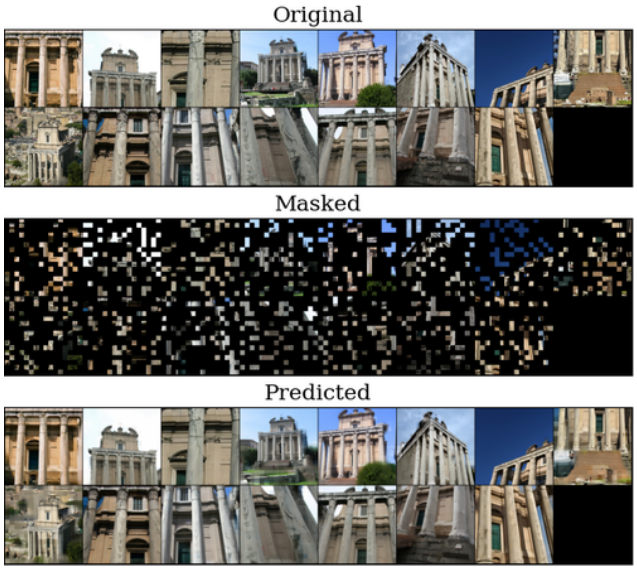


Figure 3. **Data example**. We illustrate a sequence sampled from the data and the predicted reconstructions. Note, as we train with a normalized pixel objective, we visualize the prediction plus the patch mean. Our data samples generally feature difficult viewpoint changes and varying co-visibility. Details about our sampling technique are provided in Appendix A.

4. Results

We evaluate MuM on a broad range of downstream tasks and compare primarily to the state-of-the-art feature encoders DINOv3 and CroCo v2. For computationally lighter evaluations, we also compare to MAE. Unless otherwise specified, we use the ViT-L backbone, which is the standard choice in recent 3D vision pipelines, *e.g.*, VGGT, RoMa [23], MapAnything, UFM [76].

We organize the evaluations by the number of input views n used at inference time, distinguishing between multi-view tasks ($n > 2$), two-view tasks ($n = 2$), and single-view tasks ($n = 1$).

For multi-view tasks ($n > 2$), we evaluate scene understanding by feedforward 3D reconstruction. We consider both the case with a frozen backbone and finetuning (Section 4.1). For two-view tasks ($n = 2$), we study dense matching (Section 4.2) and relative pose estimation (Section 4.3). For single-view tasks ($n = 1$), we assess monocular depth estimation (Section 4.4), surface normal estimation (Section 4.5), and semantic recognition tasks such as image classification and segmentation (Section 4.6). Finally, we conduct an ablation study on the objective and design choices (Section 4.7).

Across all multi-view 3D tasks ($n > 1$), MuM consistently outperforms DINOv3 and CroCo v2. Notably, it surpasses DINOv3 despite using substantially less compute and training data. On monocular tasks such as image clas-

sification and segmentation, which favor stronger semantic inductive biases, MuM performs below DINOv3 but remains superior to CroCo v2. Code and weights will be released on GitHub.

4.1. Multi-view tasks ($n > 2$): Feedforward 3D reconstruction

We investigate 3D understanding by multi-view feedforward reconstruction. The task is to train a neural network to estimate the 3D attributes of a scene, including camera parameters, point maps, and depth maps, from an arbitrary number of views.

Frozen encoder. Inspired by VGGT and MapAnything, we design an evaluation protocol where we train a multi-view ViT on-top of a frozen encoder. Following Wang et al. [62], we employ a ViT camera head and separate DPT [42] heads for depths and points. The model has around 700M parameters whereof around half are trainable. We train for 100K steps on BlendedMVS [72], CO3Dv2 [43], Hypersim [44], MegaDepth [34], MVS-Synth [30], PointOdyssey [77], ScanNet [15], VKitti [9], and WildRGBD [69] randomly sampling between 2 and 24 frames per scene. Each training run takes approximately 3 days on 32×H200. More details can be found in Appendix B.

Following Wang et al. [62], we evaluate multi-view camera pose estimation on Re10K [81] and CO3Dv2 [43] and point cloud estimation on DTU [32] and ETH3D [47]. In addition, we evaluate on MegaDepth [34]. The evaluated scenes have not been seen during training. The results show the usefulness of our frozen features for 3D reconstruction. MuM improves performance compared to DINOv3 and CroCo v2 on five different datasets and two different tasks, showing its versatility on both indoor, outdoor and object centric scenes.

Distillation finetuning. The full MuM architecture enables rapid finetuning on multi-view tasks, similar to how CroCo v2 can be adapted to binocular tasks. To illustrate this, we keep the decoder and simply append camera, depth and point heads to the full model. We train by distilling VGGT on our pretraining data using an unweighted L2 loss:

$$\mathcal{L}(\theta) = \sum_{i}^{n} \|\mathcal{P}_t - \mathcal{P}_s\|^2 + \|\mathcal{C}_t - \mathcal{C}_s\|^2 + \|\mathcal{D}_t - \mathcal{D}_s\|^2 \quad (2)$$

where we sum the differences between the student and teacher over the frames and $\mathcal{P}$, $\mathcal{C}$, and $\mathcal{D}$ represent world points, camera parameters, and depth maps, respectively.

We train for 100K steps using a learning rate of 1×10^{-4}, taking roughly 3 days on 16×H200. More details can be found in Appendix B. As shown in Table 2 and 3, MuM provides significant advantages over random (rand.) initialization.

Table 2. **Multi-view camera pose estimation.** Evaluating AUC@30 (↑) for 10 random frames.

Method	CO3Dv2	Re10K	MegaDepth
Frozen encoder			
CroCo v2	58.2	27.7	60.7
DINOv3	66.9	36.7	59.3
MuM	**71.5**	**50.8**	**73.0**
Distillation finetuning			
Rand.	10.6	28.0	41.8
MuM	**62.6**	**50.9**	**78.6**

Table 3. **Pointcloud estimation**. Reporting median accuracy and completeness on DTU and ETH3D.

Method	DTU		ETH3D	
	Acc. ↓	Comp. ↓	Acc. ↓	Comp. ↓
Frozen encoder				
CroCo v2	8.5	12.3	0.9	1.0
DINOv3	6.4	3.7	0.9	1.0
MuM	**3.7**	**1.6**	**0.8**	**0.8**
Distillation finetuning				
Rand.	8.4	11.7	1.0	1.4
MuM	**6.4**	**2.6**	**0.6**	**0.5**

4.2. Two-view tasks ($n = 2$): Matching

We now study frozen features for image matching. We perform both a lightweight evaluation with a linear probe and full RoMa training [23].

Linear probe. We train a linear probe on-top of the frozen features followed by a kernel nearest neighbor matcher. We evaluate average end-point-error (EPE) and robustness ($\mathcal{R}$), where robustness is defined as the share of matches with an error less than 32 pixels. In Table 4, we perform a comprehensive comparison of a wide range of vision backbones. We find that MuM produces the best representations for dense image matching. In particular, MuM outperforms both CroCo v2 and DINOv3. Notably, the performance of DINOv2 improves significantly after being fine-tuned in VGGT, indicating the need for a geometric objective. We found the results to be stable regardless of whether layer normalization is included. In Table 13 in the Appendix, we further evaluate matching without probing and find degraded performance for pixel-level objectives.

Table 4. **Comparison on dense feature matching** through linear probing. End-point-error (EPE) and robustness ($\mathcal{R}$) on MegaDepth-1500 [34, 52] and ScanNet-1500 [15, 46].

Method	Arch.	MegaDepth		ScanNet	
		EPE ↓	$\mathcal{R}$ ↑	EPE ↓	$\mathcal{R}$ ↑
Weakly-supervised					
SigLIP 2 [58]	ViT-G/16	62.1	42.3	75.8	34.6
PEcore [7]	ViT-G/14	75.8	28.8	93.6	23.2
Instance discrimination					
iBOT [80]	ViT-L/16	28.5	73.7	38.6	66.1
DINOv2 [40]	ViT-L/14	26.5	77.0	43.9	60.7
DINOv2[1]	ViT-L/14	19.3	84.6	29.7	74.4
DINOv3 [50]	ViT-L/16	19.0	86.4	28.7	78.0
DINOv3 [50]	ViT-G/16	17.1	87.2	**27.3**	77.2
Masked image modeling					
CAPI [16]	ViT-L/14	21.4	82.2	32.9	71.9
I-JEPA [2]	ViT-H/14	42.6	58.4	74.7	36.0
V-JEPA 2 [3]	ViT-H/16	60.6	37.8	86.5	23.0
MAE [29]	ViT-L/16	29.7	73.4	35.0	71.6
CroCo v2 [67]	ViT-L/16	27.3	75.7	39.0	66.6
CroCo v2[2]	ViT-L/16	22.0	80.9	29.0	76.5
MuM$^{3}_{32\times A100}$	ViT-L/16	12.0	93.7	30.2	76.0
MuM$_{64\times A100}$	ViT-L/16	**10.2**	**94.2**	27.9	**78.9**

[1]Weights when finetuned in VGGT [62].
[2]Weights when finetuned in DUSt3R [64].
[3]A lighter training run using 66 A100 days vs. 96 of CroCo v2.

Table 5. **Comparison on RoMa.** Dense matching measured in 100-percentage correct keypoints (PCK) (lower is better).

Method ↓	100-PCK@ →	1px	3px	5px
MegaDepth-1500 [34, 53]				
RoMa (DINOv2)†		13.3	4.6	3.1
CroCo v2		13.0	4.6	3.2
DINOv3		13.8	5.2	3.8
MuM		**12.5**	**4.0**	**2.6**
ScanNet-1500 [15, 46]				
RoMa (DINOv2)†		92.1	60.4	38.3
CroCo v2		92.3	61.6	40.0
DINOv3		92.3	61.0	38.9
MuM		**92.2**	**60.5**	**38.1**

†Not trained by us and patch size 14 instead of 16.

RoMa. RoMa is a state-of-the-art dense matcher achieving robust results under challenging viewpoint conditions. The model utilizes a frozen coarse encoder followed by a match decoder and convolutional refiners. We simply replace the frozen coarse encoder in RoMa and compare different backbones. We train on MegaDepth using the official pipeline, taking approximately 5 days on 4×A100-80GB. We also evaluate the outdoor model on the indoor dataset ScanNet. We report the performance in Table 5. MuM outperforms DINOv3 as well as CroCo v2 on both datasets.

4.3. Two-view tasks ($n = 2$): Relative pose

We now consider the task of relative pose estimation between two cameras through finetuning. Inspired by Dong et al. [19], we extract image tokens through a siamese encoder and the tokens are then jointly processed through cross-attention in the decoder. The features are then fed to a relative camera pose regression network and a motion averaging module and trained through a translation and rotation loss. We use a randomly initialized ViT for the decoder and compare different encoders. We jointly train the encoder and decoder on pairs extracted from BlendedMVS, CO3Dv2, DL3DV10K, MegaDepth, MVS-Synth, RealEstate10K, ScanNet, VKitti, and WildRGBD and report the results in Table 6. Training takes roughly a day on 32×A100. Further details can be found in Appendix B.

The task gives CroCo v2 a slight structural advantage, as its decoder also uses binocular cross-attention and pairs are sampled similarly to its training data. Nevertheless, MuM performs comparably or better and outperforms DINOv3.

4.4. Single-view tasks ($n = 1$): Depth estimation

We now consider the task of monocular depth estimation through linear probing on top of the frozen features. We adopt the evaluation protocol of Oquab et al. [40] and evaluate on NYUd [39] and KITTI [60], reporting the performance in Table 7. We find that MuM outperforms CroCo v2 and MAE but achieves worse results than DINOv3.

4.5. Single-view tasks ($n = 1$): Surface normals

We follow Probe3D [24] to evaluate surface normals with a DPT probe trained for 10 epochs on NYUv2 [39]. We report the performance on the test set in Table 8. As with monocular depth estimation, MuM outperforms CroCo v2 and MAE but trails DINOv3.

4.6. Single-view tasks ($n = 1$): Semantic recognition

While our objective is to create a strong feature encoder for 3D vision tasks, we evaluate its performance on the high-level semantic tasks of image classification on ImageNet-1K and semantic segmentation on ADE20K [78, 79]. We closely follow the evaluation protocol of Darcet et al. [16]. For classification, we use an *attentive probe* [2]. For segmentation, we use two lightweight classifiers on top of frozen local features, logistic regression and k-NN. We get robust results by doing an extensive sweep over hyperparameters and report the results in Table 9. DINOv3 leads on semantic tasks due to its instance segmentation

Table 6. **Two-view relative pose estimation.** Comparing AUC@ for different encoders through finetuning.

Method	MegaDepth ↑			Re10K ↑			BlendedMVS ↑		
	5°	10°	20°	5°	10°	20°	5°	10°	20°
CroCo v2	13.9	30.0	48.0	8.6	24.0	**44.9**	**7.7**	18.3	32.1
DINOv3	15.6	32.5	51.2	7.3	20.7	39.6	3.3	10.3	23.0
MuM	**26.7**	**47.0**	**65.0**	**10.3**	**25.3**	44.5	6.6	**18.4**	**35.8**

Table 7. **Depth estimation with frozen features.** We report performance when training a linear classifier on top of the frozen features. We report the RMSE metric on the 2 datasets.

Method	NYUd ↓	KITTI ↓
MAE	0.59	4.19
CroCo v2	0.51	4.09
DINOv3	**0.34**	**2.63**
MuM	0.41	3.89

Table 8. **Surface normal estimation**. We estimate surface normals with a DPT probe, following Probe3D [24], and report percentage recall at different angular thresholds and RMSE.

Method	11.25° ↑	22.5° ↑	30° ↑	RMSE ↓
MAE	49.1	72.0	79.8	26.1
CroCo v2	50.0	71.7	79.3	26.8
DINOv3	**60.4**	**79.3**	**85.4**	**22.5**
MuM	54.3	75.8	82.6	24.5

(DINO) loss, which encourages semantic understanding. MuM surpasses CroCo v2 and MAE in semantic segmentation, though for classification, it slightly lags behind MAE that is exclusively trained on ImageNet-1K.

4.7. Ablation studies

SSL objective. We verify the choice of extending MAE to multi-view by comparing to the three most popular SSL objectives for 3D vision: DINOv2, CroCo v2 and MAE. We also considered CAPI, but found the large batch size to be prohibitive. We make a comparison in an equal data and training setting by training a ViT-B for 100K steps on MegaDepth with 128 images per GPU. We report the linear probing matching performance and training time in Table 10.

We attempt to reformulate the monocular objectives in a multi-view context by randomly sampling between 2 and 12 images with alternating attention blocks providing inter-frame communication. The change constrains batch sampling to include multiple frames from the same scene and introduces information sharing between those frames, effectively reducing the variation in the data and limiting the number of scenes seen during training. Yet, we find that, while only providing a minute increase to the DINOv2 objective, it boosts MAE performance significantly. The MAE objective is substantially simpler, both conceptually and computationally, showcased through more than 3×speedup in training. We also experimented with other reconstruction targets than pixels but, while improving semantic performance, we experienced worse geometric performance.

Table 9. **Semantic tasks.** We train an attentive probe on frozen features for classification (IN1K, accuracy), and evaluate semantic segmentation (ADE20K, mIoU) using logistic regression and k-NN. The performance of the pixel reconstruction objectives lag behind the highly semantic DINOv2 and v3 models.

Method	IN1K	ADE20K	
	Att.	Lin.	k-NN
Latent prediction			
CAPI	82.9	34.4	29.2
DINOv2	85.4	39.0	34.8
DINOv3	**86.9**	**43.1**	**41.5**
Pixel reconstruction			
MAE	76.6	27.6	21.5
CroCo v2	57.4	25.0	18.8
MuM	70.8	30.2	26.0

Table 10. **SSL objective ablation.** We evaluate different objectives by training each on MegaDepth for 100K steps and report linear probing matching performance. Reformulating MAE to multiple views is fast and gives robust matches. Time denotes the number of hours of training on an 8×A100 node.

Method	Time ↓	EPE ↓
DINOv2†	38h	28.9
w/ multi-view	38h	28.4
CroCo v2	**12h**	41.4
MAE	**12h**	18.7
w/ multi-view	**12h**	**12.5**

†We use the SimDINOv2 objective [68] for multi-view stability.

	EPE ↓	Acc. ↑
2, 6	12.8	47.4
2, 12	12.5	47.4
2, 24	**10.6**	**47.8**

(a) **Sequence length.** Longer sequence lengths improve matching performance.

	EPE ↓	Acc. ↑
65%	13.3	**49.1**
75%	**10.6**	47.8
85%	12.7	45.6

(b) **Mask ratio.** The same masking ratio as MAE (75%) works well.

	EPE ↓	Acc. ↑
w/ ref.	11.9	47.7
w/o ref.	**10.6**	**47.8**

(c) **Reference view.** An unmasked reference view is not beneficial.

	EPE ↓	Acc. ↑
Encoder	16.7	43.6
Decoder	**10.6**	**47.8**

(d) **Frame communication.** Alternating attention works when used in the decoder.

	EPE ↓	Acc. ↑
Absolute	12.1	46.9
RoPE	**10.6**	**47.8**

(e) **Positional embedding**. A modern axial RoPE is superior to learned embeddings.

	EPE ↓	Acc. ↑
w/o norm	13.4	45.9
w/ norm	**10.6**	**47.8**

(f) **Reconstruction objective.** A normalized pixel objective is effective.

Table 11. **Ablations.** We train ViT-B/16 on MegaDepth for 100K steps on 8×A100 with 128 frames per GPU and report the linear probing performance on dense matching on MegaDepth and image classification on ImageNet-1K. Default settings are marked in gray.

Compared to a simple single-view MAE objective, CroCo v2 performs poorly. This can also be seen from Table 4, where MAE achieves similar results to CroCo v2 on MegaDepth and ScanNet while the latter has trained on similar data. As CroCo v2 uses a higher masking ratio (90%), we tried training on our sampled pairs (generally harder) and the pairs provided in the original paper. We find the latter to produce better results and report those.

Architecture. To verify the design of our multi-view architecture, we ablate the main design choices in Table 11. We find that many previously motivated architectural improvements are also relevant in this setting. Such as the 75% masking ratio and normalized objective used by MAE, RoPE and inter-frame communication in the decoder used by CroCo v2 and uniform sampling between 2 and 24 frames used by VGGT. In contrast, an unmasked reference view, as is used in CroCo v2, slightly degrades performance while complicating the architecture.

Feature layers. Finally, we validate our choice of extracting feature representations from the final layer throughout the preceding sections. We use ViT-L with 24 layers and report linear probing dense matching performance in Figure 4 after extracting features from different layers. We find that performance improves with deeper features.

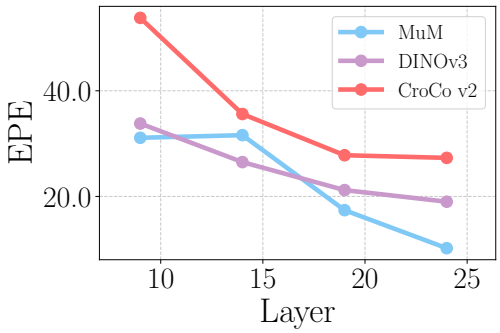


Figure 4. **Feature layer**. We report EPE for linear probing on MegaDepth. The later layers provide better representations.

5. Limitations

In Section 4.7, we found the simple pixel reconstruction loss to outperform baselines such as DINOv2 for training on multi-view data, in particular per GPU hour. Our findings there do not preclude the possibility of future research on self-distillation training on multi-view data to prove fruitful and we consider it an interesting direction to try to combine the semantic performance of DINO with geometric information from multi-view data. Moreover, due to computational constraints, we are unable to scale pretraining further than presented in the paper. Similarly, we do not have the resources to fully replicate the training and hence performance of recent feedforward reconstruction methods such as VGGT and MapAnything. However, we believe that our results provide a strong indication that our model would perform well if incorporated into such 3D vision pipelines and hope to inspire further research on the topic.

6. Conclusions

We introduce a multi-view masked-image-modeling objective that extends MAE to an arbitrary number of views of the same scene. The objective enables learning meaningful features for 3D tasks in a self-supervised manner. We showcase its effectiveness by training a ViT-L and evaluating the resulting model, MuM, on downstream tasks such as feedforward reconstruction, feature matching, and relative pose estimation. While trained using substantially less compute and data than DINOv3, MuM gives improved results when used as a feature extractor in 3D vision tasks.

Acknowledgements

This work was supported by the Wallenberg Artificial Intelligence, Autonomous Systems and Software Program (WASP), funded by the Knut and Alice Wallenberg Foundation, and by the strategic research environment ELLIIT, funded by the Swedish government. The computational resources were provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS) at C3SE, partially funded by the Swedish Research Council through grant agreement no. 2022-06725, and by the Berzelius resource, provided by the Knut and Alice Wallenberg Foundation at the National Supercomputer Centre.

References

[1] Honggyu An, Jin Hyeon Kim, Seonghoon Park, Jaewoo Jung, Jisang Han, Sunghwan Hong, and Seungryong Kim. Cross-view completion models are zero-shot correspondence estimators. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1103–1115, 2025. 2, 3

[2] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15619–15629, 2023. 6

[3] Mahmoud Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Abha Gejji, Ada Martin, Francois Robert Hogan, Daniel Dugas, Piotr Bojanowski, Vasil Khalidov, Patrick Labatut, Francisco Massa, Marc Szafraniec, Kapil Krishnakumar, Yong Li, Xiaodong Ma, Sarath Chandar, Franziska Meier, Yann LeCun, Michael Rabbat, and Nicolas Ballas. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025. 6

[4] Roman Bachmann, David Mizrahi, Andrei Atanov, and Amir Zamir. Multimae: Multi-modal multi-task masked autoencoders, 2022. 2

[5] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers, 2022. 3

[6] Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, and Elad Shulman. ARKitscenes - a diverse real-world dataset for 3d indoor scene understanding using mobile RGB-d data. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021. 4

[7] Daniel Bolya, Po-Yao Huang, Peize Sun, Jang Hyun Cho, Andrea Madotto, Chen Wei, Tengyu Ma, Jiale Zhi, Jathushan Rajasegaran, Hanoona Rasheed, Junke Wang, Marco Monteiro, Hu Xu, Shiyu Dong, Nikhila Ravi, Daniel Li, Piotr Dollár, and Christoph Feichtenhofer. Perception encoder: The best visual embeddings are not at the output of the network. *arXiv:2504.13181*, 2025. 6

[8] Lucas Brynte, José Iglesias, Carl Olsson, and Fredrik Kahl. Learning structure-from-motion with graph attention networks. *IEEE Conference on Computer Vision and Pattern Recognition*, 2024. 2

[9] Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual kitti 2, 2020. 5

[10] Yohann Cabon, Lucas Stoffl, Leonid Antsfeld, Gabriela Csurka, Boris Chidlovskii, Jerome Revaud, and Vincent Leroy. Must3r: Multi-view network for stereo 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1050–1060, 2025. 2

[11] Yohann Cabon, Lucas Stoffl, Leonid Antsfeld, Gabriela Csurka, Boris Chidlovskii, Jerome Revaud, and Vincent Leroy. Must3r: Multi-view network for stereo 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1050–1060, 2025. 2

[12] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers, 2021. 1, 2

[13] João Carreira, Dilara Gokay, Michael King, Chuhan Zhang, Ignacio Rocco, Aravindh Mahendran, Thomas Albert Keck, Joseph Heyward, Skanda Koppula, Etienne Pot, Goker Erdogan, Yana Hasson, Yi Yang, Klaus Greff, Guillaume Le Moing, Sjoerd van Steenkiste, Daniel Zoran, Drew A. Hudson, Pedro Vélez, Luisa Polanía, Luke Friedman, Chris Duvarney, Ross Goroshin, Kelsey Allen, Jacob Walker, Rishabh Kabra, Eric Aboussouan, Jennifer Sun, Thomas Kipf, Carl Doersch, Viorica Pătrăucean, Dima Damen, Pauline Luc, Mehdi S. M. Sajjadi, and Andrew Zisserman. Scaling 4d representations, 2025. 2

[14] MMCV Contributors. MMCV: OpenMMLab computer vision foundation. https://github.com/open-mmlab/mmcv, 2018. 2

[15] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 5, 6

[16] Timothée Darcet, Federico Baldassarre, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Cluster and predict latent patches for improved masked image modeling, 2025. 2, 3, 6

[17] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 4

[18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. 2

[19] Siyan Dong, Shuzhe Wang, Shaohui Liu, Lulu Cai, Qingnan Fan, Juho Kannala, and Yanchao Yang. Reloc3r: Large-scale training of relative camera pose regression for generalizable, fast, and accurate visual localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16739–16752, 2025. 2, 6

[20] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 2, 3

[21] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-Net: A Trainable CNN for Joint Detection and Description of Local Features. In *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. 1

[22] Johan Edstedt, Ioannis Athanasiadis, Mårten Wadenbäck, and Michael Felsberg. Dkm: Dense kernelized feature matching for geometry estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17765–17775, 2023. 2

[23] Johan Edstedt, Qiyu Sun, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. RoMa: Robust Dense Feature Matching. *IEEE Conference on Computer Vision and Pattern Recognition*, 2024. 1, 2, 4, 5

[24] Mohamed El Banani, Amit Raj, Kevis-Kokitsi Maninis, Abhishek Kar, Yuanzhen Li, Michael Rubinstein, Deqing Sun, Leonidas Guibas, Justin Johnson, and Varun Jampani. Probing the 3d awareness of visual foundation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21795–21806, 2024. 6, 7, 3

[25] Alaaeldin El-Nouby, Michal Klein, Shuangfei Zhai, Miguel Angel Bautista, Alexander Toshev, Vaishaal Shankar, Joshua M Susskind, and Armand Joulin. Scalable pre-training of large autoregressive image models. In *Int. Conf. Machine Learning*, 2024. 2

[26] Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva: Exploring the limits of masked visual representation learning at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19358–19369, 2023. 3

[27] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour, 2018. 4, 1

[28] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2 edition, 2004. 2

[29] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16000–16009, 2022. 1, 2, 6

[30] Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. Deepmvs: Learning multi-view stereopsis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 5

[31] Varun Jampani, Kevis-Kokitsi Maninis, Andreas Engelhardt, Arjun Karpur, Karen Truong, Kyle Sargent, Stefan Popov, Andre Araujo, Ricardo Martin-Brualla, Kaushal Patel, Daniel Vlasic, Vittorio Ferrari, Ameesh Makadia, Ce Liu, Yuanzhen Li, and Howard Zhou. Navi: Category-agnostic image collections with high-quality 3d shape and pose annotations. In *NeurIPS*, 2023. 3

[32] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engin Tola, and Henrik Aanaes. Large scale multi-view stereopsis evaluation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 5

[33] Nikhil Keetha, Norman Müller, Johannes Schönberger, Lorenzo Porzi, Yuchen Zhang, Tobias Fischer, Arno Knapitsch, Duncan Zauss, Ethan Weber, Nelson Antunes, Jonathon Luiten, Manuel Lopez-Antequera, Samuel Rota Bulò, Christian Richardt, Deva Ramanan, Sebastian Scherer, and Peter Kontschieder. MapAnything: Universal feed-forward metric 3D reconstruction, 2025. arXiv preprint arXiv:2509.13414. 2

[34] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Computer Vision and Pattern Recognition (CVPR)*, 2018. 4, 5, 6

[35] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. Lightglue: Local feature matching at light speed. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 17627–17638, 2023. 2

[36] Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22160–22169, 2024. 4

[37] Thibaut Loiseau, Guillaume Bourmaud, and Vincent Lepetit. Alligat0r: Pre-training through co-visibility segmentation for relative camera pose regression, 2025. 1

[38] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2017. 4, 2

[39] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 6

[40] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. 1, 2, 6

[41] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsuper-

vised multitask learners. *OpenAI*, 2019. Accessed: 2024-11-15. 2

[42] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12179–12188, 2021. 5

[43] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *International Conference on Computer Vision*, 2021. 4, 5

[44] Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M. Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *International Conference on Computer Vision (ICCV) 2021*, 2021. 4, 5

[45] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015. 4

[46] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947, 2020. 6

[47] Thomas Schops, Johannes L. Schonberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 5

[48] Younggyo Seo, Junsu Kim, Stephen James, Kimin Lee, Jinwoo Shin, and Pieter Abbeel. Multi-view masked world models for visual robotic manipulation. In *Proceedings of the 40th International Conference on Machine Learning*, pages 30613–30632. PMLR, 2023. 2

[49] Ketul Shah, Robert Crandall, Jie Xu, Peng Zhou, Marian George, Mayank Bansal, and Rama Chellappa. Mv2mae: Multi-view video masked autoencoders, 2024. 2

[50] Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3, 2025. 1, 2, 3, 4, 6

[51] Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding, 2023. 3

[52] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. *CoRR*, abs/2104.00680, 2021. 2, 6

[53] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: Detector-free local feature matching with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8922–8931, 2021. 6

[54] Zhenggang Tang, Yuchen Fan, Dilin Wang, Hongyu Xu, Rakesh Ranjan, Alexander Schwing, and Zhicheng Yan. Mv-dust3r+: Single-stage scene reconstruction from sparse views in 2 seconds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5283–5293, 2025. 2

[55] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017. 2

[56] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In *Advances in Neural Information Processing Systems*, 2022. 2

[57] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023. 2

[58] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier Hénaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features, 2025. 6

[59] Joseph Tung, Gene Chou, Ruojin Cai, Guandao Yang, Kai Zhang, Gordon Wetzstein, Bharath Hariharan, and Noah Snavely. Megascenes: Scene-level view synthesis at scale. In *ECCV*, 2024. 4

[60] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant cnns. In *International Conference on 3D Vision (3DV)*, 2017. 6

[61] Hengyi Wang and Lourdes Agapito. 3d reconstruction with spatial memory, 2024. 2

[62] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer, 2025. 1, 2, 5, 6

[63] Limin Wang, Bingkun Huang, Zhiyu Zhao, Zhan Tong, Yinan He, Yi Wang, Yali Wang, and Yu Qiao. Videomae v2: Scaling video masked autoencoders with dual masking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14549–14560, 2023. 2

[64] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy, 2024. 2, 4, 6

[65] Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Scalable permutation-equivariant visual geometry learning, 2025. 3

[66] Philippe Weinzaepfel, Vincent Leroy, Thomas Lucas, Romain Brégier, Yohann Cabon, Vaibhav Arora, Leonid Antsfeld, Boris Chidlovskii, Gabriela Csurka, and Jérôme Revaud. Croco: Self-supervised pre-training for 3d vision tasks by cross-view completion. *Advances in Neural Information Processing Systems*, 35:3502–3516, 2022. 1, 2

[67] Philippe Weinzaepfel, Thomas Lucas, Vincent Leroy, Yohann Cabon, Vaibhav Arora, Romain Brégier, Gabriela Csurka, Leonid Antsfeld, Boris Chidlovskii, and Jérôme Revaud. Croco v2: Improved cross-view completion pre-training for stereo matching and optical flow. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17969–17980, 2023. 2, 6

[68] Ziyang Wu, Jingyuan Zhang, Druv Pai, Xudong Wang, Chandan Singh, Jianwei Yang, Jianfeng Gao, and Yi Ma. Simplifying dino via coding rate regularization. In *Int. Conf. Machine Learning*, 2025. 7

[69] Hongchi Xia, Yang Fu, Sifei Liu, and Xiaolong Wang. Rgbd objects in the wild: Scaling real-world 3d object learning from rgb-d videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 4, 5

[70] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9653–9663, 2022. 2

[71] Jianing Yang, Alexander Sax, Kevin J. Liang, Mikael Henaff, Hao Tang, Ang Cao, Joyce Chai, Franziska Meier, and Matt Feiszli. Fast3r: Towards 3d reconstruction of 1000+ images in one forward pass. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21924–21935, 2025. 2

[72] Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. *Computer Vision and Pattern Recognition (CVPR)*, 2020. 4, 5

[73] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2023. 4

[74] Vlardimir Yugay, Duy-Kien Nguyen, Theo Gevers, Cees G. M. Snoek, and Martin R. Oswald. Visual odometry with transformers, 2025. 2

[75] Amir R Zamir, Tilman Wekel, Pulkit Agrawal, Colin Wei, Jitendra Malik, and Silvio Savarese. Generic 3D representation via pose estimation and matching. In *European Conference on Computer Vision*, pages 535–553. Springer, 2016. 4

[76] Yuchen Zhang, Nikhil Keetha, Chenwei Lyu, Bhuvan Jhamb, Yutian Chen, Yuheng Qiu, Jay Karhade, Shreyas Jha, Yaoyu Hu, Deva Ramanan, Sebastian Scherer, and Wenshan Wang. Ufm: A simple path towards unified dense correspondence with flow. In *neurips*, 2025. 4

[77] 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 *ICCV*, 2023. 5

[78] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 6

[79] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127(3):302–321, 2019. 6

[80] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *International Conference on Learning Representations (ICLR)*, 2022. 2, 6

[81] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 37, 2018. 4, 5

MuM: Multi-View Masked Image Modeling for 3D Vision

Supplementary Material

A. Further Details on Pretraining

Data sequences. Below, we outline the selection of sequences of frames. In the following section, we detail how to sample batches from sequences. For information on the dataset mixture, we refer to Section 3.4 of the main manuscript. Our protocol for finding sequences is simple, as strict co-visibility is not required. This greatly simplifies things and allows us to, without much effort, scale to around 20 million individual frames. MegaDepth and MegaScenes are the only datasets where we employ a more complicated sampling scheme. This is because each scene of the dataset is highly diverse and has no natural temporal sequencing.

For the former, we use the image overlap scores computed by D2-Net [21] and greedily sample 1,000 sequences per scene by randomly selecting an anchor frame and then chaining together images based on positive overlap. For the latter, we compute the IoU heuristic similar to CroCo and iteratively select the next image that maximizes 3D point overlap with the current image while maintaining sufficient camera viewpoint diversity. Starting from images with the most visible 3D points, we greedily build sequences until no suitable next image can be found, discarding sequences shorter than 24.

For datasets where each scene is small (*e.g.*, CO3D, DL3DV10K, Hypersim, WildRGBD), we simply select all frames in the scene as sequences and randomly sample from those during training. For the remaining datasets (3DStreetView, RealEstate10K, ScanNet++, BlendedMVS, and ARKitScenes), we use simple heuristics to form sequences. Since the frames per scene are numerically ordered, where adjacent frames are also spatially adjacent in the scene, we chunk them. We find that a chunk size of 100 frames usually works fine. After forming sequences of frames, we now detail how to select the frames in a batch.

Sampling. For each batch, independently on each GPU, we either (i) sample a full batch from ImageNet-1K (10% probability) or (ii) sample multi-view data (90% probability). To sample multi-view data, we uniformly sample a sequence length S between 2 and 24 frames. Given the sequence length, we then iteratively sample datasets with the weighting given in Table 1 and uniformly select a sequence therein. See Figure 5 for qualitative examples of sequences. We stop when the total number of frames per GPU is about to exceed 96. This gives us a batch shape of $(\text{floor}(\frac{96}{S}), S, 3, 256, 256)$. This is done independently per GPU. We sometimes refer to 96 as the batch size and use this in the linear scaling rule for learning rate [27].

Figure 5. **Data examples**. We visualize random sampled sequences from the dataset. Sequence lengths vary.

Architecture. The total model is made up of a ViT-L encoder (width 1024, depth 24 and 16 heads) followed by a ViT-B multi-view decoder (width 768, depth 12 and 12 heads). This is the exact same size as the final CroCo v2 model and thus also DUSt3R and MASt3R (excluding heads). Half of the Transformer blocks in the multi-view decoder have a global attention window and half only attend to patches within the same frame. The first block is a frame-wise attention block. The total number of trainable parameters in the model is 389.5 million, whereof around 302.3 million come from the encoder. The final layer, after the multi-view decoder, is a linear layer $W \in \mathbb{R}^{768\times768}$. The dimensions can be understood from the way that it maps each patch between the 768 embedding dimensions of the Transformer decoder to the $3 \times 16 \times 16 = 768$ color values in a patch. The loss is then computed by comparing the output of this linear mapping to the normalized (mean subtracted and divided by the standard deviation) pixel values in the ground truth patch.

Compute Resources. In Table 12, we detail the computing resources used for the main experiments. Failed or discarded training runs are excluded. The evaluations not included required only a linear probe or a k-NN classifier. These were run on a single A100 and took no longer than a couple of hours. The reporting is conducted towards academic transparency in the compute resources required to replicate our results.

B. Further Details on Experiments

Feedforward 3D reconstruction. We take inspiration from the design of VGGT and use the same heads, excluding the tracking head. Namely, we use DPT heads for world points and depths and an iterative ViT for the camera head.

Table 12. **GPU hours.** We report the compute used for the main experiments. We omit linear probing and k-NN validations.

Experiment	Model	GPUs	Time
Pretraining	MuM[†]	32×A100	3 days
Pretraining	MuM	64×A100	3 days
RoMa	MuM	4×A100	5 days
RoMa	DINOv3	4×A100	5 days
RoMa	CroCov2	4×A100	5 days
Rel. pose	MuM	32×A100	1.5 days
Rel. pose	DINOv3	32×A100	1.5 days
Rel. pose	CroCov2	32×A100	1.5 days
3D recon. (frozen)	MuM	32×H200	3 days
3D recon. (frozen)	DINOv3	32×H200	3 days
3D recon. (frozen)	CroCov2	32×H200	3 days
3D recon. (finetune)	MuM	16×H200	3 days
3D recon. (finetune)	Rand.	16×H200	3 days

[†]A lighter training run using 66 A100 days vs. 96 of CroCov2.

For the frozen encoder experiments, the points and depth heads also output confidence maps, we train for 100K steps with a learning rate of 2×10^{-4} on 32×H200 for around 3 days, and sample data following VGGT, namely, using 48 frames per GPU and an image resolution of 512. We use the AdamW [38] optimizer and minimize the same training loss as in VGGT:

$$\mathcal{L}(\theta) = \mathcal{L}_{\text{camera}} + \mathcal{L}_{\text{depth}} + \mathcal{L}_{\text{pmap}} \tag{3}$$

Where the camera loss is given by $\mathcal{L}_{\text{camera}} = \sum_{i=1}^{N} ||\hat{c}_i - c_i||_{\mathcal{H}}$ where c_i are the ground truth camera parameters and $\hat{c}_i$ are the predictions in frame i and $|| \cdot ||_{\mathcal{H}}$ denotes the Huber loss. The depth objective, $\mathcal{L}_{\text{depth}}$, follows DUSt3R and uses an aleatoric-uncertainty loss. We further incorporate the gradient-based term used in VGGT. The pointmap loss, $\mathcal{L}_{\text{pmap}}$, is similarly defined.

The finetuning distillation protocol is slightly lighter using half the batch size (due to training on half the number of GPUs) and with a learning rate of 1×10^{-4}. The distillation loss is simply the sum over the L2 errors (given in Equation 2). We train without confidence weights.

Dense matching. For the linear probing evaluation we take as basis the MegaDepth1500 and ScanNet1500 dense evaluation protocols in RoMa and DKM. We train a square linear projection on top of the frozen encoder features followed by a log-softmax kernel nearest neighbor matcher to estimate the dense warp. We achieve a robust metric by sweeping over learning rates and weight decays in parallel. We train for 25K steps with batch size 8 and evaluate every 2,500 steps, reporting the best performance achieved. The evaluation protocol takes roughly an hour on an A100.

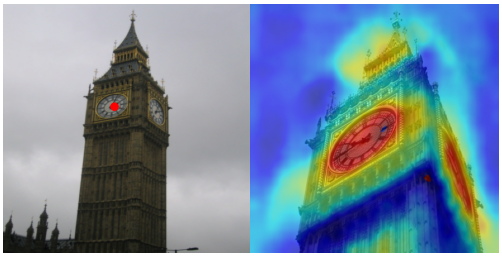


Figure 6. **Attention map for query patch**. We visualize the global attention map in the decoder for a query patch. We find that the attention score is highest for matching patches.

For RoMa, we follow their exact training setup. which is the following. Training for 3 million steps on 560×560 resolution with a frozen encoder. This takes roughly 5 days on $4\times$A100-80GB. We opt to only train on MegaDepth (commonly referred to as outdoor) and use this model for all evaluations.

Two-view relative pose estimation. The decoder architecture follows exactly Reloc3r [19] but is not initialized from DUSt3R but instead trained from scratch. We train the decoder and the heads with a learning rate of 2×10^{-4} and finetune the encoder with a learning rate of 2×10^{-5}. We use a total batch size of 1024 and train on 16×A100 for approximately 1 day.

Monocular depth estimation. We follow the protocol in DINOv2 [40]. Namely, we use the MMCV [14] package and apply random cropping, rotation and flipping augmentation. We crop the images to 480×640 and treat it as a classification problem with 256 bins. We use a combination of cross entropy and gradient loss. We train with AdamW, cosine annealing and a peak learning rate of 1×10^{-4}.

Semantic recognition. For both classification and segmentation, we follow exactly the protocol in CAPI [16]. In particular, the classification protocol uses an attentive probe. This is due to MuM, CroCo v2 and MAE, not having a meaningful global representation in the same way as the DINO family (following its instance discrimination objective on the CLS token). The attentive probe is trained in a supervised fashion and its output is processed by a trained linear layer.

C. Additional experiments

Qualitative examples. Inspired by ZeroCo [1], we visualize the global attention map between a query patch in the first image and all the patches in the second image. The result, illustrated in Figure 6, clearly illustrates that the attention map contains information about correspondences be-

Table 13. **Cosine similarity feature matching**. We report average recall on NAVI [31] following the protocol of Probe3D [24] and EPE on MegaDepth and ScanNet.

Method	NAVI ↑	MegaDepth ↓	ScanNet ↓
MAE	43.7	88.8	83.4
CroCo v2	41.0	60.3	62.1
DINOv3	**62.8**	**36.5**	56.4
MuM	46.9	50.8	**55.7**

tween the two frames. An interesting future direction is to investigate zero-shot matching directly from the attention map, similar to ZeroCo [1].

We also visualize the dense warp estimated through linear probing in Section 4.2 in Figure 7.

Cosine similarity feature matching. In our main results, we report the linear probing dense feature matching accuracy. However, prior work has explored the matchability of the raw features. While El Banani et al. [24] notes that more expressive probes are necessary to determine 3D understanding, they opt for correspondences based on cosine similarity in their paper. We follow their protocol and report the matching performance in Table 13. As suggested by Siméoni et al. [50], we evaluate both with and without layer normalization on the frozen features and report the best result. Interestingly, generative objectives such as MuM, MAE, and CroCo v2 perform worse when evaluated without a linear probe. This might be related to how the DINO-family uses a patch similarity objective whereas the generative objectives tested work in pixel space. We further find that MuM outperforms CroCo v2 and MAE and matches DINOv3 on ScanNet.

Figure 7. **Dense warp**. Warp estimated through linear probing on MegaDepth on top of frozen features.