In Pursuit of Pixel Supervision for Visual Pre-training

Supplementary Material

1. Failure Attempts, Limitations, and Future Directions

1.1. Failure Attempts

In addition to the three aforementioned modifications to MAE, we explored several other approaches that ultimately did not yield performance improvements, including but not limited to:

- **Multi-block masking:** We experimented with both inpainting (predicting center regions given surrounding context) [1] and outpainting variants (predicting surrounding context given center regions) [5]. Compared to our adopted masking strategy that is based on $n \times n$ local patches, these approaches introduce additional hyper-parameter complexity, requiring careful tuning of the number of blocks, block scale ranges, and block aspect ratios. Furthermore, they constrain the diversity of masking patterns available during training. For instance, the outpainting variant consistently provides a large contiguous region of visible context, which limits the model's ability to learn long-range dependencies across spatially distant patches. Empirically, neither variant delivered performance gains over our simpler patch-block masking approach.
- **Hybrid masking ratios:** MAE, Pixio, as well as many other masked image modeling works employ a fixed masking ratio across training. However, different images may benefit from different masking ratios depending on their complexity. For simple images with high redundancy, aggressive masking ratios are necessary to create a sufficiently challenging pretext task. Conversely, for complex images with rich, non-redundant content, excessively high masking ratios can make reconstruction unpredictable, causing the model to converge to trivial solutions rather than learning meaningful representations. To address this, several works [10, 6, 18] have proposed adaptive mechanisms that dynamically determine optimal masks based on motion cues or attention maps. However, these approaches introduce additional complexity and may exhibit bias toward specific image distributions (*e.g.*, object-centric datasets). We explored a simpler alternative: hybrid masking ratios. For each training image, we randomly sample a masking ratio from a pre-defined set (*e.g.*, [62.5%, 75%, 87.5%]), allowing both simple and complex images to be trained with more appropriate difficulty levels. While this design seems conceptually reasonable, we did not observe clear improvements.
- **Hybrid masking granularity:** As shown in Table 7, we observe that different downstream tasks benefit from different masking granularity during pre-training. Mid-level, geometry-focused tasks (*e.g.*, depth estimation) perform better with finer masking granularity (*e.g.*, 2×2 patch blocks), while high-level, semantics-oriented tasks (*e.g.*, semantic segmentation) favor coarser masking granularity (*e.g.*, 4×4 patch blocks). This is expected, as easier, smaller masking units encourage the model to capture fine-grained spatial relationships, whereas harder, larger masking units promote learning of broader semantic context. Given these complementary benefits, a natural strategy is to employ hybrid masking granularity during training. We randomly vary the masking block size across batches to help the model adapt to different contextual scales and develop multi-level visual understanding. However, despite extensive attempts, we found that using a single, fixed masking granularity throughout training consistently yields the best performance.
- **Koleo loss on class tokens:** DINOv2 [14] employs Koleo loss [16] to enforce uniformity in the feature distribution across samples, encouraging informative representations. However, this constitutes a strong manually-imposed inductive bias, as semantically similar samples may naturally have similar representations and should not be artificially repelled. We explored an alternative application: applying Koleo loss to our multiple class tokens rather than to individual samples. The motivation is to encourage each class token to capture distinct aspects of the image (*e.g.*, semantics, scene layout, lighting, style), thereby promoting functional specialization among tokens. In preliminary experiments, this regularization yielded minor improvements on dense prediction tasks. However, it severely degraded ImageNet classification performance with these class tokens. More critically, we observed training instability even with very small loss weights (*e.g.*, 0.1). Given these issues, we ultimately excluded this regularization from our framework.
- **Cross-attention in decoder:** We observed that in both the original MAE and our Pixio, reconstructed masked regions exhibit higher visual quality than reconstructed visible regions. This occurs because the reconstruction loss is only computed on masked tokens, leading to optimization bias toward these tokens. This phenomenon also implies that the appended learnable `[MASK]` tokens and the encoder-extracted visible tokens reside in different feature spaces. Given this observation, we hypothesized that employing cross-attention between visible and masked tokens [7], rather than full self-attention,

might better model their distinct representations while facilitating information transfer. Although this modification provided marginal computational speedup by reducing the attention complexity, it did not yield improvements in downstream performance. We therefore retained the standard self-attention mechanism in our final design.

- **Predicting both masked and visible patches.** Following the aforementioned observation of misaligned feature spaces between mask tokens and visible tokens, we explored applying reconstruction loss to both masked and visible patches. In practice, this substantially degraded model performance across downstream tasks. This finding demonstrates that plain autoencoding on all image patches is suboptimal for learning transferable visual representations, and that the asymmetric reconstruction objective is crucial to MAE's effectiveness.
- **Predicting partial masked patches:** MAE reconstructs all masked patches at the decoder. However, masked patches themselves contain redundancy. Neighboring masked patches often have similar content. Also considering the increased computational cost of our deeper decoder, we attempted to reconstruct only a randomly sampled subset of masked patches [7], thereby reducing decoder overhead while maintaining the pretext task. However, while this provided marginal training speedup, it consistently degraded downstream performance.
- **Feeding multi-stage features to the decoder:** MAE uses only the encoder's final block output for decoding. Since pixel reconstruction requires both high-level semantic understanding and low-level textural details, relying solely on the final features may place excessive burden on the last encoder block, potentially compromising its ability to learn high-level abstractions. Ideally, different encoder stages should naturally capture different levels of visual information. Motivated by this, we extracted intermediate features from four encoder stages, concatenated them along the channel dimension, and fed this fused representation to the decoder. Our hypothesis was that combining early-stage and late-stage features would enable natural complementarity under pixel-level supervision. However, improvements were marginal and inconsistent across tasks. Therefore, we retained the simpler single-stage design in our final framework.

In summary, our three presented modifications (*i.e.*, deeper decoder, larger masking blocks, and additional class tokens) represent minimal yet critical improvements to MAE. We highly value such simplicity in design. While some above explored alternatives may indeed be viable, we were unable to identify optimal configurations that consistently improved performance. We hope these empirical insights will inform future research in masked image modeling.

In addition to pre-training framework, we have also tried other data curation strategies, including but not limited to:

- **Online hard example mining:** Rather than pre-computing image difficulty using a pre-trained MAE model on uncurated data, we explored selecting informative samples dynamically during training. Specifically, at each training iteration, we performed a forward pass on a batch of N candidate images and computed reconstruction loss for each. We then backpropagated only through the k images with the highest reconstruction loss, where $k = \alpha \cdot N$ and $\alpha \in (0, 1]$ controls the selection ratio. However, this approach proved problematic in practice. Early in training, when the model has not yet learned meaningful representations, the loss-based difficulty estimation is unreliable and noisy. This instability can lead to suboptimal convergence, as the training distribution shifts unpredictably. Therefore, we adopted the offline pre-computation strategy instead, which provides more stable difficulty estimates.
- **Canny edge density as a proxy for sample difficulty:** In addition to reconstruction loss, we explored using Canny edge density [2], which is measured as the proportion of edge pixels detected in an image, as a heuristic proxy for image complexity. The intuition is that images with richer edge structures may contain more informative visual content. However, we found that such hand-crafted edge detectors are overly sensitive to low-level patterns and repetitive textures (*e.g.*, grass, fabric patterns, brick walls), which produce high edge responses but offer limited semantic diversity.

1.2. Limitations and Future Directions

We recognize fundamental limitations in applying *masking to static images*. On the positive side, masking constructs a meaningful pretext task. High masking ratios encourage models to learn both high-level semantics and low-level details. Nevertheless, random masking remains an *artificial distortion*, which introduces undesirable biases. In practice, masking presents unavoidable trade-offs. Low masking ratios cause ground truth leakage, making reconstruction trivial. High masking ratios provide insufficient context for learning and create distribution shift between training and inference. Critically, masked image modeling never exposes the model to natural, complete images during training. Despite these limitations, masking (or other artificial distortions) appears necessary for image-based pre-training.

The fundamental reason why so many artificial distortions and human inductive biases are necessary is, *static images* have inherent limitations as a medium for learning visual intelligence. Images are not the natural format in which visual

information exists in the physical world. Humans do not learn from isolated snapshots. Instead, we learn through continuous temporal experiences in a causal manner, observing how the world evolves over time. From this perspective, *video* deserves greater emphasis, particularly long videos that capture the natural progression of events and their causal relationships. Videos offer a crucial advantage: the temporal dimension enables natural predictive objectives without artificial masking. Models can predict future frames from current observations—a task grounded in the causal structure of the physical world. This eliminates the need for artificial spatial masking [19] or noise injection [4].

This work serves as a pioneering validation that pixel in data *alone* can produce strong visual representations competitive with more complex pre-training paradigms. Looking forward, we will scale this supervision approach to web-scale video data. By leveraging the temporal richness of videos and natural predictive objectives, we aim to develop more powerful and less biased visual foundation models for both videos and images.

2. Implementation Details

2.1. Pre-training

The basic hyperparameters for our pre-training closely follow the original MAE framework, with several adaptations for large-scale training. Given our web-scale training data, we extend the training iterations from 500K to 1.3M and increase the batch size from 4,096 to 16,384. Importantly, we find that reducing the peak learning rate (2.4e-3 $\rightarrow$ 8e-4) is essential for stable convergence on less curated, more diverse web data. We also increase the input resolution from 224×224 to 256×256 with a patch size of 16×16. Comprehensive pre-training configurations are detailed in Table 1, and the complete architecture details of our largest Pixio-5B model are provided in Table 2.

config	value
data	2B web-crawled images
iterations	1,284,000
batch size	16,384
input resolution	256×256
precision	bfloat16
optimizer	AdamW
learning rate	8e-4
learning rate schedule	cosine decay
warmup steps	128,400
weight decay	0.05
optimizer momentum	β_1, β_2 = 0.9, 0.95
data augmentation	RandomResizedCrop
crop scale	0.2-1.0
crop ratio	0.75-1.33
drop path	0.4
masking ratio	75%
masking granularity	4×4 patches

Table 1. Pre-training details.

config	value
encoder	
#params	5.4B
patch size	16×16
#blocks	48
embedding dimension	3072
hidden dimension	12288
attention heads	32
positional embedding	learnable
#class tokens	8
decoder	
#params	103M
#blocks	32
embedding dimension	512
hidden dimension	2048
attention heads	16
positional embedding	learnable

Table 2. Teacher model details.

2.2. Distillation

Using our pre-trained Pixio-5B encoder as the teacher, we distill a series of smaller, more efficient student encoders: Pixio-1B (1.4B parameters), Pixio-H (631M), Pixio-L (303M), and Pixio-B (86M). Specifically, the teacher encoder processes unmasked images while the student encoder receives either masked or the same unmasked inputs. For capable students (*e.g.*, Pixio-1B, Pixio-H), we use masked inputs. This encourages the student to learn robust representations despite partial information. We align student features to teacher features through a lightweight MLP projection head, optimizing cosine similarity loss at both patch-token and class-token levels. The two losses are equally weighted and averaged for optimization. We use 50% masking ratio with 4×4 patch masking granularity. All students are trained for 500K iterations with batch size 8,192 and learning rate 1e-3. We use drop path [9] 0.4 for students Pixio-1B and Pixio-H, while using drop path 0.1 for less capable students Pixio-L and Pixio-B. All other hyper-parameters remain identical to the pre-training stage.

2.3. Downstream Evaluation

We have open-sourced the downstream evaluation code to facilitate reproducibility. We highlight some details below.

ImageNet-1K classification. For k-NN protocol, we follow DINO's [3] official implementation. We report (k-NN) accuracy with $k = 10$. For fine-tuning protocol, we follow MAE's official implementation. In both cases, images are resized to 256 pixels on the shorter side and then center-cropped to 256×256 for inference. For Pixio, we average all the class tokens to obtain the global representation.

Monocular depth estimation and semantic segmentation. We evaluate under two settings: a trainable DPT head [15] or a linear regression/classification head, with the encoder frozen in both cases. Following DINOv2 [14], we find that for certain encoders (*e.g.*, DINOv2, MAE, Pixio), concatenating patch tokens with (averaged) class tokens along the channel dimension yields better performance than using patch tokens alone. We therefore report results using the optimal configuration for each encoder. For the DPT head, we extract intermediate features evenly from four encoder stages. All models are trained for 60 epochs. To ensure fair comparison across architectures, we use training resolution 256×256 for encoders with patch size 16 and 224×224 for those with patch size 14, maintaining consistent effective sequence length. During inference, we apply a sliding window approach with overlap and ensemble the predictions from overlapping regions.

We report all encoders' results under our evaluation protocol for its simplicity. It is worth noting that our protocol is different from that used in DINOv3 paper. For clarity, we borrow DINOv3 officially reported results in Table 3. The protocol differences (at least) include: 1) we use a linear regression head for depth estimation, while DINOv3 uses a classification head with bins, 2) we use resolution 256×256 for downstream fine-tuning, while DINOv3 uses 416×544 for depth estimation and 512×512 for semantic segmentation, and 3) DINOv3 uses test-time augmentation for depth estimation, whereas we do not.

For Depth Anything [20], MapAnything [11], and CortexBench [12], we follow the official implementations, replacing the encoder with our pre-trained models while keeping all other components unchanged.

	DINOv3 Protocol		Our Protocol	
Model	NYUv2 Linear Head	ADE20K Linear Head	NYUv2 Linear Regression Head	ADE20K Linear Head
DINOv3-H+	0.352	54.8	0.559	50.3

Table 3. DINOv3 officially reported results.

3. Ablation Studies

Limited by space in the main paper, we primarily presented ablation results through figures. Here, we provide detailed numerical results and more comprehensive ablation configurations. We also provide additional ablation studies on the pre-training data and supervision signal.

3.1. Block-Wise Performance of MAE and Pixio

Table 4 shows the block-wise feature quality of the official MAE models. Notably, the best generic features reside far before the final encoder block. For instance, on ADE20K semantic segmentation with MAE-H, there is a substantial 3.3 mIoU performance gap between the last encoder block and the optimal intermediate block, indicating that the final layers sacrifice representation quality for reconstruction. However, with our deeper decoder design, this issue is largely resolved. As shown in Table 5, Pixio's final encoder block produces competitive features, with only a negligible 0.006 RMSE gap on NYUv2 compared to the best intermediate block. This validates our hypothesis that insufficient decoder capacity forces MAE's late encoder blocks to assume decoding responsibilities.

3.2. Decoder Design

Beyond the decoder widths (768, 512, 384 dimensions) presented in the main paper, Table 6 additionally reports results with an even shallower decoder (256 dimensions). The results confirm that excessively shallow decoders are suboptimal, as they lack sufficient capacity for the challenging pixel reconstruction objective.

3.3. Masking Design

Extending the analysis in the main paper, Table 7 provides a comprehensive evaluation of different masking configurations (varying both masking ratio and granularity) under the 384×32 decoder setting. These results further validate the importance

ViT	Block Index	IN-1K KNN ↑	NYUv2 RMSE ↓	KITTI RMSE ↓	ADE20K mIoU ↑	Pascal mIoU ↑
H/14	32	55.0	0.593	4.411	35.1	70.7
	28	60.1	0.583	4.247	36.3	71.9
	24	**62.1**	0.574	4.298	37.4	73.6
	20	61.9	**0.564**	**4.218**	**38.4**	**74.4**
	16	51.1	0.612	4.465	37.3	73.6
L/16	24	57.7	0.585	4.607	34.2	70.7
	21	**61.1**	0.585	4.451	35.4	72.1
	18	60.0	**0.582**	**4.285**	**36.2**	**73.0**
	15	40.7	0.622	4.433	34.3	70.8
	12	28.4	0.711	4.864	28.4	62.2

Table 4. Probing officially released MAE-H/14 (1280×32) and MAE-L/16 (1024 ×24) encoders [8], which are trained on ImageNet-1K. Decoder: 512×8. We use a linear head for both monocular depth estimation (regression) and semantic segmentation (classification).

Block Index	IN-1K KNN ↑	NYUv2 RMSE ↓	KITTI RMSE ↓	ADE20K mIoU ↑	Pascal mIoU ↑
48	68.4	0.360	3.603	50.2	82.0
42	69.7	**0.354**	3.583	50.3	**82.4**
36	70.4	0.361	**3.570**	**50.7**	81.9
30	**70.8**	0.370	3.579	50.3	82.0
24	70.2	0.390	3.575	49.8	81.6

Table 5. Probing our Pixio-5.4B encoder (3072×48), which is trained from scratch on our curated 2B images. Decoder: 512×32. We use a linear head for both monocular depth estimation (regression) and semantic segmentation (classification).

Decoder		IN-1K KNN ↑	NYUv2 RMSE ↓	KITTI RMSE ↓	ADE20K mIoU ↑	Pascal mIoU ↑
Width	Depth					
768	8	44.2	0.480	3.156	35.4	71.9
	16	58.3	0.408	2.828	41.3	77.1
	32	49.0	0.458	3.007	36.7	75.1
	48	32.0	0.574	3.418	26.7	62.8
512	8 (MAE)	35.3	0.431	2.986	35.8	71.6
	16	55.1	0.409	2.789	39.5	76.1
	32	55.8	0.410	2.749	40.4	76.9
	48	57.6	0.422	2.832	40.5	77.1
384	8	35.2	0.469	3.047	32.1	68.3
	16	48.6	0.425	2.825	36.6	73.3
	32	56.2	0.410	2.821	39.7	75.2
	48	55.6	0.412	2.940	39.8	76.8
256	8	32.6	0.499	2.995	29.1	64.8
	16	38.6	0.473	3.001	32.4	68.5
	32	47.2	0.451	2.923	35.7	71.2
	48	43.7	0.437	2.898	37.4	74.4

Table 6. Ablation study on the decoder on a ViT-H encoder (1280-dim×32-blocks). Here we mask at a single patch and use 1 class token.

of larger masking granularity for learning better representations.

We also sweep more masking ratios on different pre-training data in Table 8. Too aggressive masking ratios (*e.g.*, 87.5%) and too conservative masking ratios (*e.g.*, 50%) are suboptimal in most cases.

Decoder	Masking		IN-1K KNN ↑	NYUv2 RMSE ↓	KITTI RMSE ↓	ADE20K mIoU ↑	Pascal mIoU ↑
	Ratio	Granularity					
512×8	75%	1×1 (MAE)	35.3	0.431	2.986	35.8	71.6
		2×2	54.3	0.362	2.653	41.8	77.1
		4×4	43.3	0.373	2.895	42.7	78.3
	62.5%	1×1	32.6	0.468	2.944	31.2	66.6
		2×2	49.5	0.378	2.715	38.6	74.2
		4×4	53.2	0.356	2.654	41.8	78.1
512×32	75%	1×1	55.8	0.410	2.749	40.4	76.9
		2×2	63.3	0.358	2.782	44.5	80.2
		4×4	63.5	0.387	2.932	43.5	79.9
	62.5%	1×1	52.2	0.444	2.896	37.6	75.2
		2×2	62.8	0.360	2.650	43.7	79.2
		4×4	52.8	0.360	2.741	44.3	79.5
384×32	75%	1×1	56.2	0.410	2.821	39.7	75.2
		2×2	61.1	0.351	2.697	43.7	79.0
		4×4	61.8	0.366	2.909	44.4	80.2
	62.5%	1×1	46.6	0.450	2.907	35.3	72.1
		2×2	57.0	0.359	2.675	42.4	78.2
		4×4	57.9	0.357	2.725	44.5	79.4

Table 7. Ablation study on masking ratio and masking granularity (measured in #patches). Here we use 1 class token.

Data	Granularity	Masking Ratio							
		50%		62.5%		75%		87.5%	
		ADE20K ↑	KITTI ↓	ADE20K	KITTI	ADE20K	KITTI	ADE20K	KITTI
ImageNet-21K	1×1	34.5	4.57	38.1	2.79	40.8	2.77	41.9	2.92
Mapillary		20.7	5.04	19.3	7.10	17.2	3.86	12.9	3.95
MetaCLIP-S		38.3	4.51	41.4	2.68	42.8	3.50	41.8	3.60
ImageNet-21K	2×2	42.4	2.67	44.2	2.62	44.8	2.75	41.7	3.09
Mapillary		14.5	3.83	12.1	4.17	13.6	3.81	14.1	3.51
MetaCLIP-S		44.3	3.71	46.1	2.60	46.8	2.58	45.5	2.82

Table 8. Sweeping more masking ratios on different pre-training data. The Mapillary [13] dataset is composed of 1.5M driving images.

3.4. Number of Class Tokens

In addition to the comparisons in the main paper, Table 9 ablates whether class tokens should be included in the decoder input. We observe that feeding class tokens to the decoder yields slightly better performance, suggesting that allowing them to participate in reconstruction helps learn more informative global representations.

3.5. Pre-training on Existing Curated Datasets

We have demonstrated the advantage of our curated MetaCLIP-S data over ImageNet-1K, ImageNet-21K, and YFCC100M data sources. We further attempt to use the existing curated dataset LAION [17] for pre-training. As compared in Table 10, our data yields better performance than LAION before and after our curation, showing the superiority of our data source and the importance of our proposed self-curation.

Decoder	#[CLS]	In Decoder	IN-1K KNN ↑	NYUv2 RMSE ↓	KITTI RMSE ↓	ADE20K mIoU ↑	Pascal mIoU ↑
512×32	1 (MAE)	✓	63.3	0.358	2.782	44.5	80.2
	4	✓	75.1	0.360	2.746	44.8	80.7
	8	✓	75.0	0.361	2.654	44.8	80.5
	16	✓	74.0	0.360	2.775	45.0	80.7
	1	✗	64.1	0.373	2.787	44.3	80.1
	4	✗	68.9	0.364	2.663	44.8	80.2
	8	✗	70.6	0.376	2.794	44.2	80.0
	16	✗	71.9	0.373	2.728	44.2	80.4
384×32	1 (MAE)	✓	61.1	0.351	2.697	43.7	79.0
	4	✓	68.9	0.350	2.683	43.9	80.2
	8	✓	70.6	0.346	2.687	44.5	80.2
	16	✓	71.0	0.352	2.736	44.4	80.0
	1	✗	62.6	0.362	2.688	43.9	79.3
	4	✗	66.0	0.369	2.784	43.3	79.7
	8	✗	68.7	0.361	2.762	44.1	79.7
	16	✗	70.6	0.356	2.715	44.3	80.0

Table 9. Ablation study on the number of class tokens and whether to include them in the decoder. Here we mask at 2×2 patch blocks.

Data	Our Curation	NYUv2 RMSE ↓	KITTI RMSE ↓	ADE20K mIoU ↑	Pascal mIoU ↑
LAION	✗	0.407	3.382	41.5	73.4
	✓	0.366	2.973	44.3	77.1
MetaCLIP	✗	0.351	3.126	44.7	78.0
MetaCLIP-S	✓	0.321	2.581	46.8	80.1

Table 10. Ablation study on using existing curated dataset for pre-training.

3.6. Whether to Apply Patch-Level Normalized Pixels

MAE [8] adopts patch-level normalized pixels instead of raw pixels as the supervision to enhance the contrast locally. We ablate this choice in Table 11 using our curated web-scale data. The larger converged training loss and better performance further convince the effectiveness of this simple practice.

Patch Normalization	Training Loss	NYUv2 RMSE ↓	KITTI RMSE ↓	ADE20K mIoU ↑	Pascal mIoU ↑
✗	0.388	0.359	2.767	45.0	78.6
✓	0.493	0.321	2.581	46.8	80.1

Table 11. Ablation study on whether to apply patch-level normalized pixels or raw pixels for supervision.

References

[1] 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 *CVPR*, 2023. 1

[2] John Canny. A computational approach to edge detection. *TPAMI*, 2009. 2

[3] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021. 4

[4] Xinlei Chen, Zhuang Liu, Saining Xie, and Kaiming He. Deconstructing denoising diffusion models for self-supervised learning. In *ICLR*, 2025. 3

[5] Timothée Darcet, Federico Baldassarre, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Cluster and predict latent patches for improved masked image modeling. *TMLR*, 2025. 1

[6] David Fan, Jue Wang, Shuai Liao, Yi Zhu, Vimal Bhat, Hector Santos-Villalobos, Rohith MV, and Xinyu Li. Motion-guided masking for spatiotemporal representation learning. In *ICCV*, 2023. 1

[7] Letian Fu, Long Lian, Renhao Wang, Baifeng Shi, Xudong Wang, Adam Yala, Trevor Darrell, Alexei A Efros, and Ken Goldberg. Rethinking patch dependence for masked autoencoders. *arXiv:2401.14391*, 2024. 1, 2

[8] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022. 5, 7

[9] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *ECCV*, 2016. 3

[10] Ioannis Kakogeorgiou, Spyros Gidaris, Bill Psomas, Yannis Avrithis, Andrei Bursuc, Konstantinos Karantzalos, and Nikos Komodakis. What to hide from your students: Attention-guided masked image modeling. In *ECCV*, 2022. 1

[11] Nikhil Keetha, Norman Müller, Johannes Schönberger, Lorenzo Porzi, Yuchen Zhang, Tobias Fischer, Arno Knapitsch, Duncan Zauss, Ethan Weber, Nelson Antunes, et al. Mapanything: Universal feed-forward metric 3d reconstruction. *arXiv:2509.13414*, 2025. 4

[12] Arjun Majumdar, Karmesh Yadav, Sergio Arnaud, Jason Ma, Claire Chen, Sneha Silwal, Aryan Jain, Vincent-Pierre Berges, Tingfan Wu, Jay Vakil, et al. Where are we in the search for an artificial visual cortex for embodied intelligence? In *NeurIPS*, 2023. 4

[13] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *ICCV*, 2017. 6

[14] 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. *TMLR*, 2024. 1, 4

[15] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *ICCV*, 2021. 4

[16] Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, and Hervé Jégou. Spreading vectors for similarity search. In *ICLR*, 2019. 1

[17] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. In *NeurIPS*, 2022. 6

[18] Jeongwoo Shin, Inseo Lee, Junho Lee, and Joonseok Lee. Self-guided masked autoencoder. In *NeurIPS*, 2024. 1

[19] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In *NeurIPS*, 2022. 3

[20] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. In *NeurIPS*, 2024. 4