MultiMAE: Multi-modal Multi-task Masked Autoencoders
Supplementary Material

Roman Bachmann*, David Mizrahi*, Andrei Atanov, and Amir Zamir

Swiss Federal Institute of Technology Lausanne (EPFL)
`{firstname.lastname}@epfl.ch`
`https://multimae.epfl.ch/`

Table of Contents

A Additional Pre-training Implementation Details

We report the default pre-training setting in Table 1. The learning rate follows the linear scaling rule [10]: $lr = base\_lr \times \text{batchsize}/256$. The number of non-masked tokens given to the encoder is set to 49 when using a single input modality (mask ratio of 3/4), and 98 when using 2 or 3 modalities (mask ratio of 3/4 and 5/6, respectively). Furthermore, given that the semantic segmentation map consists of 64-dimensional class embeddings, naively projecting each patch to a token is computationally expensive (when flattened, each patch would have a dimension of 16384 and the projection layer would have approx. 12M parameters). To make this projection efficient while keeping the number of segmentation patches constant, we downsample the semantic segmentation input by a factor of 4 and use patches of size 4×4.

MultiMAE Decoder. We illustrate the MultiMAE decoder in Fig 1. Following MAE [12], each decoder has a linear projection layer to adapt the outputs from the encoder to the decoder dimension. After this linear projection, we add both sine-cosine positional embeddings and learned modality embeddings to the decoder inputs. This is then followed by a cross-attention layer, a MLP, and two Transformer blocks.

---

[*] Equal contribution

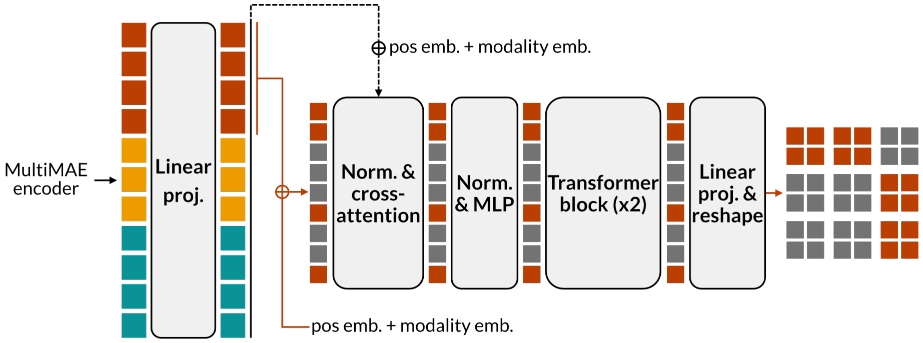

Fig. 1: MultiMAE decoders: Tokens from the MultiMAE encoder are first linearly projected to the decoder dimension, after which positional and modality-specific embeddings are added. A cross-attention step integrates information from tokens of other modalities before applying an MLP and two Transformer blocks. Finally, each token is projected and reshaped to form an image. In this illustration, each token expands into four pixels.

B Transfer Implementation Details

B.1 ImageNet Classification Fine-tuning Setting

For ImageNet-1K [8] classification, we follow the end-to-end fine-tuning procedure from MAE [12] and replace the decoders by an average pooling operation over all encoded tokens, followed by LayerNorm [2] and a linear projection. The default setting is shown in Table 2.

B.2 Semantic Segmentation

The typical approach [3, 12] to fine-tuning Vision Transformers for semantic segmentation is not suited for multi-modal inputs in two aspects: 1) the segmentation head and 2) the evaluation procedure. We cover these two aspects next and propose a simplified fine-tuning setting for semantic segmentation to overcome these issues.

Segmentation Head. The UPerNet [31] head used in BEiT [3] and MAE [12] operates on a feature pyramid [19]. While a Vision Transformer operating only on RGB images can be modified to return hierarchical feature maps through the use of deconvolution layers on intermediate features [35], this procedure is not so simple when the input is multi-modal. In contrast, segmentation heads that operate *only* on the output tokens do not have this issue. One such head is the Segmenter [28], for which tokens are passed through additional Transformer blocks, then reshaped into a feature map and upsampled to full resolution. However, the direct upsampling can result in inprecise segmentation maps and hurt performance. Instead, we propose using a simple segmentation head based on the ConvNeXt architecture [20]. First, we increase the dimensionality D of the

Table 1: Default pre-training setting. For ablations, the number of epochs is set to 400. For best results, the number of epochs is set to 1600

Hyperparameters	Value
Optimizer	AdamW [22]
Base learning rate [10]	1e-4
Weight decay	0.05
Adam β	(0.9, 0.95)
Batch size	2048
Learning rate sched.	Cosine decay [21]
Training epochs	400 or 1600
Warmup learning rate	1e-6
Warmup epochs	40
Non-masked tokens	98
Sampling α	1.0
Task weighting	None (equal weights)
Input resolution	224 × 224
Augmentation	RandomResizedCrop

Table 2: ImageNet-1K classification setting. We follow the fine-tuning settings from MAE [12]

Hyperparameters	Value
Optimizer	AdamW [22]
Base learning rate [10]	5e-4
Weight decay	0.05
Adam β	(0.9, 0.999)
Layer-wise lr decay [7]	0.65
Batch size	1024
Learning rate sched.	Cosine decay [21]
Training epochs	100
Warmup learning rate	1e-6
Warmup epochs	5
Input resolution	224 × 224
Augmentation	RandAugment(9, 0.5)
Label smoothing	0.1
Mixup [34]	0.8
Cutmix [32]	1.0
Drop path [16]	0.1

Table 3: Comparison of semantic segmentation heads. We report the mIoU (↑) on ADE20K [36], Hypersim [26] and NYUv2 [27]. The proposed segmentation head based on the ConvNeXt [20] architecture performs on average slightly better than Segmenter [28]

Method	Head	ADE20K	Hypersim	NYUv2
MultiMAE	Segmenter-Mask [28]	**46.3**	36.0	49.0
MultiMAE	ConvNeXt	46.2	**37.0**	**52.0**

output tokens wih a linear projection, and then reshape the tokens to form a feature map of size $H/4 \times W/4 \times D/8$. We then apply 4 ConvNeXt blocks on this feature map before upsampling it to full resolution using bilinear interpolation. We find that this simple ConvNeXt head outperforms Segmenter, as shown in Table 3. To adapt this head to multi-modal inputs, we can either select only the output tokens from a single modality (as information from other modalities gets passed to these tokens through self-attention) or concatenate tokens from different modalities. We find that both approaches perform comparably and select the former as it is slightly more efficient.

Evaluation Procedure. Vision Transformers are commonly evaluated using the sliding window procedure from MMSegmentation [23] (e.g., [4,35,37]). This procedure involves first resizing the validation images so that the *smallest side* matches the training resolution[*], and then applying a sliding window over the resized image and averaging predictions across windows. However, this procedure is not suitable if the input modalities rely on statistics from the entire image (e.g.,

---

[*] In some implementations, the height is resized to the training resolution, which most often coincides with the smallest side.

Table 4: Semantic segmentation fine-tuning settings for ADE20K [36], Hypersim [26] and NYUv2 [27]

Hyperparameters	ADE20K	Hypersim	NYUv2
Optimizer	AdamW [22]		
Learning rate	1e-4		
Layer-wise lr decay [7]	0.75		
Weight decay	0.05		
Adam β	(0.9, 0.999)		
Batch size	16	16	8
Learning rate sched.	Cosine decay [21]		
Training epochs	64	25	200
Warmup learning rate	1e-6		
Warmup epochs	1		
Input resolution	512 × 512	512 × 512	640 × 640
Augmentation	Large scale jittering (LSJ) [9]		
Color jitter	✓		
Drop path [16]	0.1		

standardized depth) or do not have a 2D structure (e.g., object bounding boxes). Therefore, we use a simpler evaluation procedure inspired by [18], which consists of resizing the image so that the *largest side* matches the training resolution and padding the smallest side. As the evaluated images have a smaller resolution, this simple procedure results in slightly worse reported performance compared to sliding windows. However, it can be used regardless of the input modalities and thus allows for a more fair comparison of segmentation performance for different modalities.

Training Details. The semantic segmentation transfer settings for all three segmentation datasets are shown in Table 4. Following [18], our main augmentation is large scale jittering (LSJ) [9]. We also apply color jittering with the following parameters: `brightness=0.4, contrast=0.4, saturation=0.2, hue=0.1, p=0.5`.

B.3 NYUv2 Depth Estimation

For depth estimation on the NYUv2 dataset. [27], we resize all images from 640 × 480 to 341 × 256. During training, we randomly crop the images to 256 × 256 and during testing, we take a central crop of size 256 × 256.

We follow [11, 17] and apply color jittering with the following parameters: `brightness=0.1255, contrast=0.4, saturation=0.5, hue=0.2, p=0.5`. We also randomly turn the image into gray-scale with probability $p = 0.3$.

We use the DPT [25] head to decode layers [3,6,9,12] of the ViT-B encoder into the dense depth map. For training, we use the reverse Huber loss [17]. Detailed transfer settings are given in Table 5. For evaluation, we measure the δ_1 metric on the test set, showing the percentage of pixels p with error $\max\{\frac{\hat{y}_p}{y_p}, \frac{y_p}{\hat{y}_p}\}$ less than 1.25.

Table 5: Fine-tuning settings for NYUv2 [27] depth estimation and eight Taskonomy [33] 2D regression tasks

Hyperparameters	NYUv2 depth	Taskonomy tasks
Optimizer	AdamW [22]	
Learning rate	1e-4	3e-4
Layer-wise lr decay [7]	0.75	
Weight decay	1e-4	5e-2
Adam β	(0.9, 0.999)	
Batch size	128	32
Learning rate sched.	Cosine decay [21]	
Training epochs	2000	100
Warmup learning rate	1e-6	
Warmup epochs	100	5
Input resolution	256 × 256	384 × 384
RandomCrop	✓	✗
Color jitter	✓	✗
Drop path [16]	✗	0.1

B.4 Taskonomy Dense Regression Tasks

We transfer to the following eight dense regression tasks from the Taskonomy [33] dataset: *Principal curvature, z-buffer depth, texture edges, occlusion edges, 2D keypoints, 3D keypoints, surface normals, and reshading.* We train the transfers on a random subset of the Taskonomy-tiny split, selecting 800 training and 200 validation images. The test evaluation is performed on the entire Taskonomy-tiny test split (54514 images), using the checkpoint with the lowest validation loss.

For training and testing, all images are resized to 384×384 and we perform no further augmentations. As for NYUv2 [27] depth estimation, we use the DPT [25] head, accessing layers [3,6,9,12] from the ViT-B encoder. All tasks are trained with an L1 loss. Detailed transfer settings are given in Table 5.

C Mask Sampling Strategies

We sample the number of non-masked tokens per modality using a Dirichlet distribution with concentration parameter $\alpha = 1$. Figure 2 illustrates the sampling behavior under different α values. For simplicity, we picked $\alpha = 1$ for all our experiments in the main paper, which exposes the models to a large diversity of masks. Samples using $\alpha = 1$ include cases where all tokens are sampled from a single modality (very low α) and MultiMAE has to fully reconstruct the other two, cases where all modalities are equally represented (very high α), and everything in between.

In Table 6, we show transfer results on ImageNet-1K [8] classification and ADE20K [36] semantic segmentation using MultiMAE models trained with $\alpha \in \{0.2, 0.5, 1.0, \infty\}$. By $\alpha = \infty$, we denote always sampling an equal number of

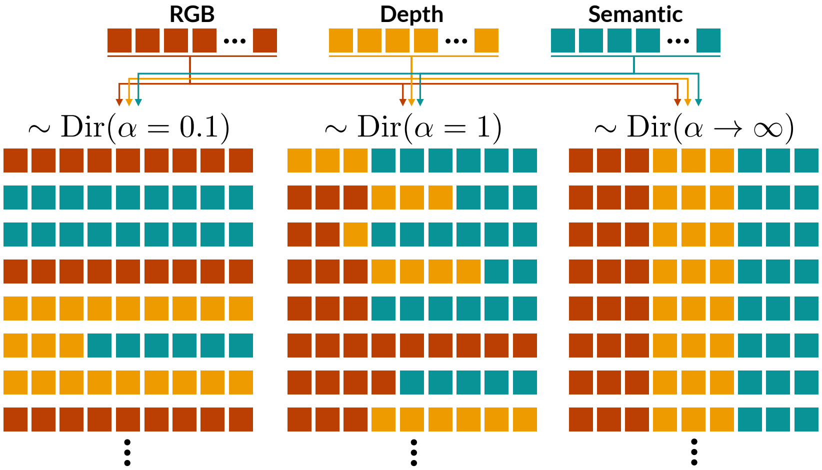
RGB
Depth
Semantic
$\sim \mathrm{Dir}(\alpha = 0.1)$
$\sim \mathrm{Dir}(\alpha = 1)$
$\sim \mathrm{Dir}(\alpha \rightarrow \infty)$

Fig. 2: Multi-modal mask sampling: We sample the proportion of tokens per modality using a symmetric Dirichlet distribution $\mathrm{Dir}(\alpha)$ with concentration parameter α. We illustrate here the sampling behavior for different choices of α values when selecting nine tokens from three modalities. Each row represents one sample of tokens. With small α, most tokens will be sampled from single modalities, while large α values result in equal representation of each modality. Setting $\alpha = 1$ is equivalent to sampling uniformly over the support and results in a more diverse sampling behavior.

Table 6: Comparison of mask sampling strategies. We report RGB-only transfers to ImageNet-1K [8] classification and ADE20K [36] semantic segmentation using MultiMAEs pre-trained with different Dirichlet concentration parameter α. All models were trained for 400 epochs and do *not* use the additional per-patch-standardized RGB decoder (see Sec. 3.4 in main paper). By $\alpha = \infty$ we denote always sampling an equal number of visible tokens for each task

α	ImageNet-1K [8]	ADE20K [36]
0.2	82.7	44.6
0.5	82.5	44.8
1.0	82.8	**45.1**
∞	**82.9**	42.9

tokens from each modality. All models in this table were trained for 400 epochs and do not include the additional per-patch-standardized RGB head (see Sec. 3.4 in the main paper). Setting $\alpha = 1$ performs best on ADE20K, while being close second on ImageNet-1K behind $\alpha = \infty$. Smaller values of α do not perform better on these two RGB-only downstream tasks, even though during training they were exposed to more samples that contain tokens from only one modality. Biasing the sampling towards modalities that will be used during transfer is an interesting future direction.

D Detailed Taskonomy Transfer Results

Table 4 in the main paper compared several baselines by their average rank on eight different Taskonomy [33] downstream tasks. In this section, we show per-task results of all these baselines. Table 7 shows detailed results for the ablation on the choice of MultiMAE pre-training tasks, while Table 8 shows results for the comparison to single-task and multi-task baselines.

Out of these eight Taskonomy tasks, the edges and 2D-keypoints task labels were originally created from RGB images, while the other tasks were rendered from the scanned scene mesh. A pre-training scheme that includes depth should thus transfer better to the depth-related tasks, such as surface normals. Indeed, we observe this in Table 7, where MultiMAE pre-trained using depth transfer better than MAE or the RGB-S MultiMAE. Importantly, additionally including semantic segmentation along RGB and depth in the pre-training does not degrade performance on these tasks.

In Table 8, we see that MultiMAE performs similarly to the single-task RGB→D baseline that was trained using full RGB inputs. For the single and multi-task baselines, the right choice of pre-training task(s) is crucial, as for example the RGB→S baselines performs consistently worse than the ones including depth, as well as the MultiMAE RGB-S baseline from Table 7.

Table 7: Taskonomy transfer results using MultiMAE models pre-trained on a **varying number of modalities**, where the pre-training modalities are the same as the target tasks. Downstream transfers are trained from RGB-only. All models were pre-trained for 400 epochs. We report L1 losses (↓) and indicate with **bold** and underline the best and second-best results, respectively

Method	Curvature ($\cdot 10^2$)	Depth ($\cdot 10^2$)	Edges ($\cdot 10^3$)	Occlusion ($\cdot 10^4$)	2D-keypoints ($\cdot 10^4$)	3D-keypoints ($\cdot 10^2$)	Normals ($\cdot 10^2$)	Reshading ($\cdot 10$)	Average loss ($\cdot 10^2$)	Average rank
MAE (D2)	4.455	3.651	4.608	6.237	2.736	4.585	6.189	1.120	3.828	3.75
RGB-D	4.249	3.378	4.031	6.608	**2.440**	4.447	6.094	1.051	3.646	2.125
RGB-S	4.276	3.406	**3.868**	5.939	2.615	4.467	6.139	1.067	3.678	2.625
RGB-D-S	**4.236**	**3.340**	5.290	**5.924**	2.590	**4.432**	**6.086**	**1.040**	**3.639**	**1.5**

Table 8: Taskonomy transfer results comparing pre-trained single-task and multi-task baselines (pre-trained using ***non-masked*** RGB-only inputs) against the RGB-D-S MultiMAE. Downstream transfers are trained from RGB-only. All models were pre-trained for 400 epochs. We report L1 losses (↓) and indicate with **bold** and underline the best and second-best results, respectively

Method	Curvature ($\cdot 10^2$)	Depth ($\cdot 10^2$)	Edges ($\cdot 10^3$)	Occlusion ($\cdot 10^4$)	2D-keypoints ($\cdot 10^4$)	3D-keypoints ($\cdot 10^2$)	Normals ($\cdot 10^2$)	Reshading ($\cdot 10$)	Average loss ($\cdot 10^2$)	Average rank
RGB→D	4.251	**3.222**	7.038	**5.914**	2.790	4.458	**5.960**	**1.013**	**3.602**	1.625
RGB→S	4.314	3.666	7.206	6.051	3.029	4.595	6.843	1.155	3.973	4
RGB→D-S	4.266	3.465	6.745	5.949	2.899	4.510	6.264	1.080	3.759	2.875
MultiMAE	**4.236**	3.340	**5.290**	5.924	**2.590**	**4.432**	6.086	1.040	3.639	**1.5**

E Robustness Evaluation on ImageNet

We study the robustness of the ImageNet [8] fine-tuned models by evaluating them on four different ImageNet-like validation sets [13–15,30] that contain various domain-shifts and corruptions, and we show the results in Table 9. To that end, we directly use the models that were fine-tuned on ImageNet-1K classification, and evaluate them without any modifications on the respective robustness evaluation datasets. MultiMAE performs better than all baselines of the same model size (ViT-B) on ImageNet-R and ImageNet-S. It also performs better than MAE on ImageNet-C, but falls behind DINO [5] and MoCo-v3 [6]. On ImageNet-A, MultiMAE performs worse than DINO and MAE, but better than the supervised and MoCo-v3 baselines.

Table 9: Robustness evaluation on ImageNet variants from **RGB-only**. We report the top-1 accuracy on the IN-1K validation split, as well as robustness evaluations on IN-Adversarial [15], IN-Corruption [14] (mean corruption error), IN-Rendition [13], as well as IN-Sketch [30]

Method	IN-1K ↑	IN-A ↑	IN-C ↓	IN-R ↑	IN-S ↑
Supervised [29]	81.8	24.2	49.7	43.5	31.4
DINO [5]	83.1	**35.5**	**45.5**	48.1	35.4
MoCo-v3 [6]	82.8	33.2	<u>46.2</u>	48.4	<u>35.6</u>
MAE [12]	**83.3**	<u>35.1</u>	51.6	<u>49.3</u>	35.5
MultiMAE	**83.3**	33.9	49.1	**50.5**	**37.1**

F Comparison of MAE Variants

In Section 4.2 in the main paper, we compare MultiMAE to a pre-trained MAE with a decoder of depth 8, following the best-performing setting described in [12]. However, as our MultiMAE uses shallower and narrower decoders, we also pre-train MAE with a decoder of similar depth (2) and width (256). We compare these two MAE versions in Table 10. We find that while these two models perform comparably on ImageNet-1K classification, as reported in [12], using a deeper decoder leads to a stark increase in performance for all other tasks. Given the benefits of a larger decoder for MAE, it stands to reason that MultiMAE could also benefit from using wider and deeper decoders, even though that would significantly increase pre-training time.

Furthermore, it has been observed that MAE models pre-trained using the official PyTorch [24] implementation (such as ours) do not exactly match the results of a MAE trained using the original (and unavailable) TensorFlow [1] implementation[†]. Therefore, we also report results using model weights from the

---

[†] A discussion about the reproducibility issues of MAE can be found at: https://github.com/facebookresearch/mae/issues/30

TensorFlow implementation to assess the impact of the codebase on transfer performance. We observe minor differences in transfer performance, with the original TensorFlow implementation slightly outperforming the PyTorch implementation on all tasks.

Table 10: Comparison of MAE Variants. We report the top-1 accuracy (↑) on ImageNet-1K [8] (IN-1K) classification (C), mIoU (↑) on ADE20K [36], Hypersim [26], and NYUv2 [27] semantic segmentation (S), as well as δ_1 accuracy (↑) on NYUv2 depth (D). Text in **bold** and <u>underline</u> indicates the first and second-best results, respectively. All models are pre-trained for 1600 epochs. D2 = Decoder of depth 2 and width 256. D8 = Decoder of depth 8 and width 512

Method	IN-1K (C)	ADE20K (S)	Hypersim (S)	NYUv2 (S)	NYUv2 (D)
MAE (D2, PyTorch)	<u>83.3</u>	43.3	34.1	46.9	83.7
MAE (D8, PyTorch)	<u>83.3</u>	<u>46.2</u>	<u>36.5</u>	<u>50.1</u>	<u>85.1</u>
MAE (D8, TensorFlow)	**83.6**	**46.5**	**37.1**	**50.9**	**85.4**

G Comparison of Pre-training Time

We report the pre-training epoch time in Table 11. By using shallow decoders, the training time of MultiMAE is comparable to MAE (with a decoder of depth 8) despite having twice the amount of unmasked tokens and multiple decoders. Note that removing masked tokens from the encoder, as proposed by MAE, is crucial in enabling pre-training on multiple dense modalities.

Table 11: Pre-training time comparison. Pre-training epoch time for MAE [12] and MultiMAE on ImageNet-1K [8]. We train with 8 Nvidia A100 GPUs and use PyTorch with automatic mixed precision enabled. D2 = Decoder of depth 2 and width 256. D8 = Decoder of depth 8 and width 512. w/ `[M]` = Mask tokens also given to the ViT-B encoder

Method	Encoder	Num. unmasked	Epoch time (mins)
MAE (D2)	ViT-B	49	2.7
MAE (D8)	ViT-B	49	5.0
MultiMAE	ViT-B	98	6.0
MultiMAE, w/ `[M]`	ViT-B	98	43.3

H Additional Visualizations

Figure 3 shows MultiMAE for varying number of visible patches and highlights the robustness of the model with respect to masking ratios far from the training mask ratio. Figure 4 shows more visualizations on ImageNet-1K [8] validation set images. For all examples, 98 visible patches were sampled using Dirichlet concentration parameter $\alpha = 1$. Figure 5 further shows predictions where we sample three random masks for each image.

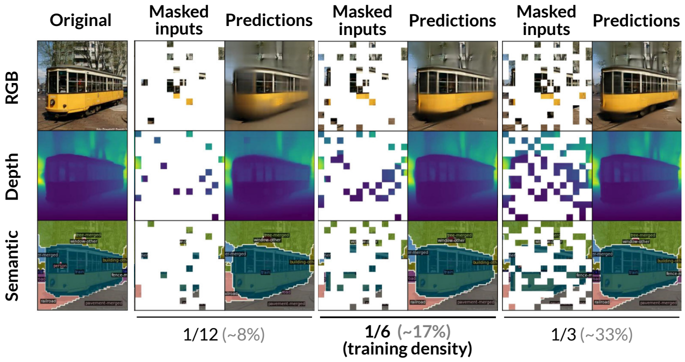

Fig. 3: MultiMAE predictions for a varying number of visible patches. The predictions are plausible even when given half the number of patches seen during pre-training, and the reconstruction quality improves as the number of visible patches increases. An interactive visualization is available on our website.

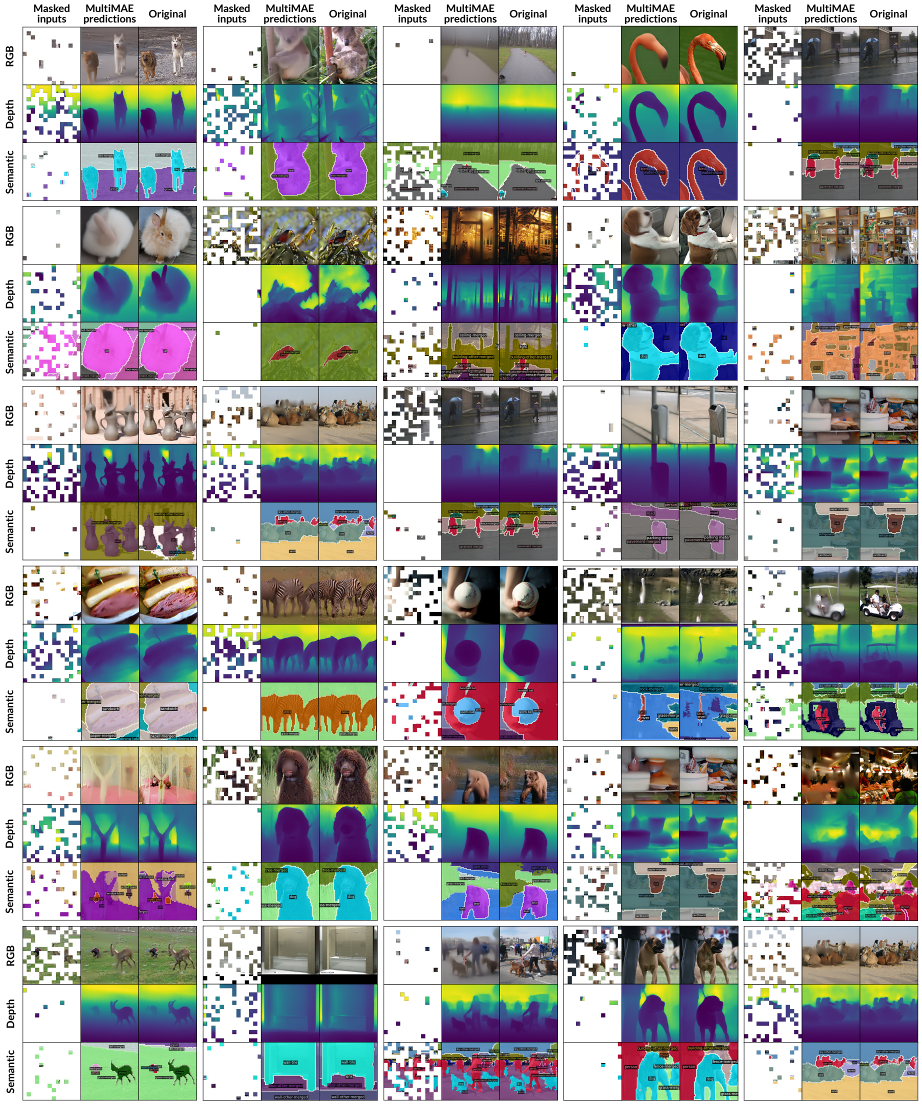

Fig. 4: MultiMAE predictions on ImageNet-1K validation set samples. 98 visible patches were sampled using Dirichlet concentration parameter $\alpha = 1$.

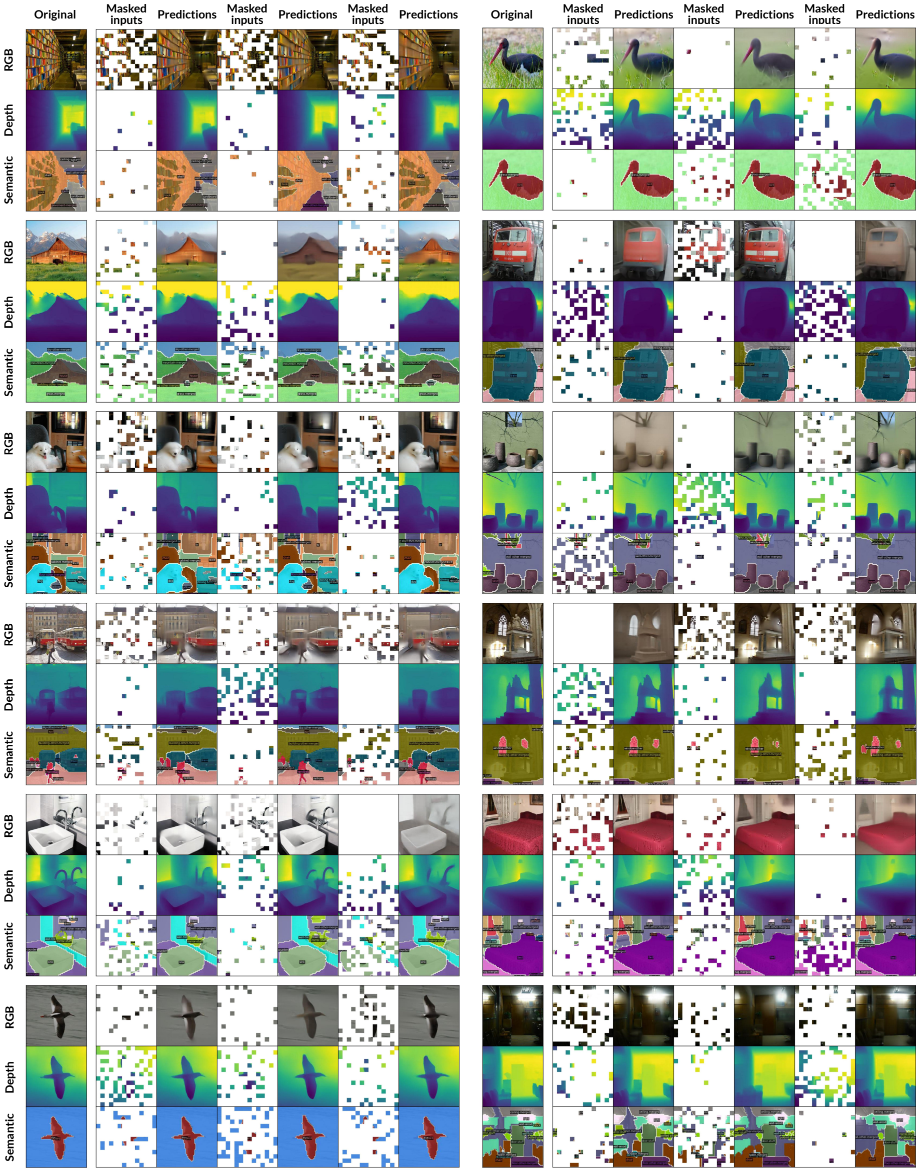

Fig. 5: MultiMAE predictions on ImageNet-1K validation set samples. 98 visible patches were sampled using Dirichlet concentration parameter $\alpha = 1$. For each image, we sample three random masks.

References

1. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X.: TensorFlow: Large-scale machine learning on heterogeneous systems (2015), `https://www.tensorflow.org/`, software available from tensorflow.org 8
2. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv preprint arXiv:1607.06450 (2016) 2
3. Bao, H., Dong, L., Wei, F.: Beit: Bert pre-training of image transformers. ArXiv **abs/2106.08254** (2021) 2
4. Bao, H., Dong, L., Wei, F.: BEiT: BERT Pre-Training of Image Transformers. arXiv:2106.08254 [cs] (Jun 2021), `http://arxiv.org/abs/2106.08254`, arXiv: 2106.08254 3
5. Caron, M., Touvron, H., Misra, I., J'egou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. 2021 IEEE/CVF International Conference on Computer Vision (ICCV) pp. 9630–9640 (2021) 8
6. Chen, X., Xie, S., He, K.: An empirical study of training self-supervised vision transformers. 2021 IEEE/CVF International Conference on Computer Vision (ICCV) pp. 9620–9629 (2021) 8
7. Clark, K., Luong, M.T., Le, Q.V., Manning, C.D.: Electra: Pre-training text encoders as discriminators rather than generators. arXiv preprint arXiv:2003.10555 (2020) 3, 4, 5
8. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR (2009) 2, 5, 6, 8, 9, 10
9. Ghiasi, G., Cui, Y., Srinivas, A., Qian, R., Lin, T.Y., Cubuk, E.D., Le, Q.V., Zoph, B.: Simple copy-paste is a strong data augmentation method for instance segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2918–2928 (2021) 4
10. Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., He, K.: Accurate, large minibatch sgd: Training imagenet in 1 hour. arXiv preprint arXiv:1706.02677 (2017) 1, 3
11. Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al.: Bootstrap your own latent-a new approach to self-supervised learning. Advances in Neural Information Processing Systems **33**, 21271–21284 (2020) 4
12. He, K., Chen, X., Xie, S., Li, Y., Doll'ar, P., Girshick, R.B.: Masked autoencoders are scalable vision learners. ArXiv **abs/2111.06377** (2021) 1, 2, 3, 8, 9
13. Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T.L., Parajuli, S., Guo, M., Song, D.X., Steinhardt, J., Gilmer, J.: The many faces of robustness: A critical analysis of out-of-distribution generalization. 2021 IEEE/CVF International Conference on Computer Vision (ICCV) pp. 8320–8329 (2021) 8
14. Hendrycks, D., Dietterich, T.G.: Benchmarking neural network robustness to common corruptions and perturbations. ArXiv **abs/1903.12261** (2019) 8
15. Hendrycks, D., Zhao, K., Basart, S., Steinhardt, J., Song, D.X.: Natural adversarial examples. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 15257–15266 (2021) 8

16. Huang, G., Sun, Y., Liu, Z., Sedra, D., Weinberger, K.Q.: Deep networks with stochastic depth. In: European conference on computer vision. pp. 646–661. Springer (2016) 3, 4, 5
17. Laina, I., Rupprecht, C., Belagiannis, V., Tombari, F., Navab, N.: Deeper depth prediction with fully convolutional residual networks. In: 2016 Fourth international conference on 3D vision (3DV). pp. 239–248. IEEE (2016) 4
18. Li, Y., Xie, S., Chen, X., Dollar, P., He, K., Girshick, R.: Benchmarking Detection Transfer Learning with Vision Transformers. arXiv:2111.11429 [cs] (Nov 2021), `http://arxiv.org/abs/2111.11429`, arXiv: 2111.11429 4
19. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2117–2125 (2017) 2
20. Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A convnet for the 2020s. arXiv preprint arXiv:2201.03545 (2022) 2, 3
21. Loshchilov, I., Hutter, F.: Sgdr: Stochastic gradient descent with warm restarts. arXiv preprint arXiv:1608.03983 (2016) 3, 4, 5
22. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: ICLR (2019) 3, 4, 5
23. MMSegmentation Contributors: OpenMMLab Semantic Segmentation Toolbox and Benchmark (7 2020), `https://github.com/open-mmlab/mmsegmentation` 3
24. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: NeurIPS (2019) 8
25. Ranftl, R., Bochkovskiy, A., Koltun, V.: Vision transformers for dense prediction. 2021 IEEE/CVF International Conference on Computer Vision (ICCV) pp. 12159–12168 (2021) 4, 5
26. Roberts, M., Paczan, N.: Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. 2021 IEEE/CVF International Conference on Computer Vision (ICCV) pp. 10892–10902 (2021) 3, 4, 9
27. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from rgbd images. In: ECCV (2012) 3, 4, 5, 9
28. Strudel, R., Garcia, R., Laptev, I., Schmid, C.: Segmenter: Transformer for semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7262–7272 (2021) 2, 3
29. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., J'egou, H.: Training data-efficient image transformers & distillation through attention. In: ICML (2021) 8
30. Wang, H., Ge, S., Xing, E.P., Lipton, Z.C.: Learning robust global representations by penalizing local predictive power. In: NeurIPS (2019) 8
31. Xiao, T., Liu, Y., Zhou, B., Jiang, Y., Sun, J.: Unified perceptual parsing for scene understanding. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 418–434 (2018) 2
32. Yun, S., Han, D., Oh, S.J., Chun, S., Choe, J., Yoo, Y.: Cutmix: Regularization strategy to train strong classifiers with localizable features. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 6023–6032 (2019) 3
33. Zamir, A.R., Sax, A., Shen, W.B., Guibas, L.J., Malik, J., Savarese, S.: Taskonomy: Disentangling task transfer learning. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (2018) 5, 7

34. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. arXiv preprint arXiv:1710.09412 (2017) 3
35. Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., Fu, Y., Feng, J., Xiang, T., Torr, P.H., et al.: Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6881–6890 (2021) 2, 3
36. Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: Scene parsing through ade20k dataset. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 5122–5130 (2017) 3, 4, 5, 6, 9
37. Zhou, J., Wei, C., Wang, H., Shen, W., Xie, C., Yuille, A., Kong, T.: ibot: Image bert pre-training with online tokenizer. arXiv preprint arXiv:2111.07832 (2021) 3