A Implementation details

In this section, we provide the implementation details of MetaMAE, including architectures and hyperparameters for MetaMAE.

Architectural details. We summarize our architectures in Table 1, with the hyperparameter notation referred from [26]. We use token embedding for encoder inputs, and apply positional embedding to both encoder and decoder inputs, as suggested by [26]. Specifically, token embedding separates the input data into fixed-size tokens, while positional embedding uses a fixed, absolute position represented by a combination of sine and cosine functions. We describe the token size for each specific dataset in Appendix C.

Table 1: A Pytorch-like architecture description of MetaMAE. $n \in \{2, 4, 6\}$, $p \in \{0, 0.1\}$ are the hyperparameters.

Component	Layer descriptions
Encoder f_θ	$\texttt{TransformerBlock}(d_{\text{model}} = 256, d_{\text{ff}} = 512, h = 8, P_{\text{drop}} = p, \text{GELU}, \text{LayerNorm=True}) \times 12$
Decoder g_ϕ	$\texttt{TransformerBlock}(d_{\text{model}} = 128, d_{\text{ff}} = 256, h = 4, P_{\text{drop}} = 0, \text{GELU}, \text{LayerNorm=True}) \times n$
Projector h_ψ	$\texttt{Linear}(256, 1028), \texttt{BatchNorm1d}(1028), \texttt{Linear}(1028, 128)$

Hyperparameter details. We summarize our selected hyperparameters for each dataset in Table 2. We observe that a certain set of hyperparameters can generally work across modalities, e.g., (α, λ, decoder depth) = (0.5, 0.1, 4), or can be shared within each modality, e.g., $P_{\text{drop}} = 0$ for Token modality. Table 3 demonstrates that MetaMAE with shared (α, λ, decoder depth) can outperform the previous results. However, we recommend specific values for each modality to improve the performance (refer to Appendix B for the hyperparameter sensitivity). We fix the temperature term for a contrastive loss $\tau = 0.5$ and the Nearby-$\mathcal{S}$ ratio $r = 0.1$. For latent adaptation, we update the latent representation using a single-step update, where the update magnitude is α. Following [25], we train for 100k iterations for pretraining, and 100 epochs for linear evaluation. For pretraining, we use the AdamW optimizer [12] with both the learning rate and weight decay set to 1e-4. For linear evaluation, we use the Adam optimizer [9], also with the learning rate and weight decay set to 1e-4. The batch size used for both pretraining and linear evaluation is as described in [24, 25].

Table 2: Hyperparameters of MetaMAE for pretrain datasets.

Modality	**Time-series**	**Tabular**	**MS Image**	**Token**		**Speech**	**RGB Image**	
Dataset	PAMAP2	HIGGS	EuroSAT	Genom	Pfam	Libri	WaferMap	ImageNet32
MetaMAE specific hyperparameters								
α	0.5	1.0	0.1	0.1	0.1	0.1	0.1	0.1
λ	1.0	1.0	1.0	0.01	1.0	1.0	0.1	0.1
decoder depth	4	6	4	2	4	4	6	4
$P_{\texttt{drop}}$	0.1	0.1	0	0	0	0	0	0
Hyperparameters from DABS benchmarks								
mask ratio	0.85	0.50	0.85	0.50	0.15	0.85	0.85	0.85
batch size	256	256	64	32	128	64	128	64

Table 3: Linear evaluation performance with the shared hyperparameters across modalities, *viz.*, of (α, λ, decoder depth) = (0.5, 0.1, 4), compared to the previous best results reported.

Modality	**Time-series**	**Tabular**	**MS Image**	**Token**		**Speech**	**RGB Image**
Dataset	PAMAP2	HIGGS	EuroSAT	Genom	Pfam	Libri	WaferMap
Previous best [25]	85.3	70.0	86.3	53.6	54.7	60.2	93.9
MetaMAE (ours)	**89.1**	**71.0**	**88.5**	**55.4**	**62.2**	**77.1**	**95.4**

B Analysis on hyperparameter sensitivity

We here provide more ablation experiments with varying hyperparameters α, λ, decoder depth, P_{drop}, and latent adaptation step size. Table 4, 5, 6, and 7 show the sensitivity of hyperparameters on the PAMAP2 and WaferMap datasets. We observe that MetaMAE performs well even with non-optimal hyperparameters, except for the decoder depth and P_{drop}, but we suggest finding better hyperparameters specific to each domain (e.g., $\lambda = 0.1$ for WaferMap). Regarding the decoder depth, we find that each modality requires an appropriate value, but generally, MetaMAE performs well with a decoder depth of 4. In Table 8, we observe that single-step adaptation effectively achieves good performance, and in some cases, even outperforms multiple-step adaptation due to the risk of overly decoder-specific support representation.

Table 4: Sensitivity of α on PAMAP2 and WaferMap.

α	PAMAP2	WaferMap
0.1	86.2	**95.5**
0.5	**89.3**	95.4
1.0	89.1	95.2

Table 5: Sensitivity of λ on PAMAP2 and WaferMap.

λ	PAMAP2	WaferMap
0.01	88.6	95.2
0.1	**89.1**	**95.5**
1.0	**89.3**	93.6

Table 6: Sensitivity of decoder depth on PAMAP2 and WaferMap.

depth	PAMAP2	WaferMap
2	84.9	94.2
4	**89.3**	**95.5**
6	86.2	**95.5**

Table 7: Sensitivity of P_{drop} on PAMAP2 and WaferMap.

P_{drop}	PAMAP2	WaferMap
0	79.4	**95.5**
0.1	**89.3**	94.7

Table 8: Sensitivity of latent adaptation step size on PAMAP2 and WaferMap.

step size	PAMAP2	WaferMap
1	89.3	**95.5**
5	**89.6**	94.9

C Dataset details

We provide a summary of the considered datasets from the DABS benchmarks [24, 25] in Table 9. Note that we use the dataset split described in [24, 25].

Table 9: Datasets considered for pretraining and linear evaluation in our experiments. "MS Image" denotes the Multi-spectral image modality.

Modality	Dataset	# of classes	Input shape	Token shape
Time-series	PAMAP2 [19]	12	52×320	5
Tabular	HIGGS [18]	2	28	1
MS Image	EuroSAT [8, 7]	10	$13 \times 64 \times 64$	8×8
Token	Genomics [20]	10	4×250	1
	Genomics-OOD [20]	60	4×250	1
	Pfam [5]	623	26×128	1
	SCOP [6]	1195	26×128	1
	Secondary Structure [10, 2]	4	26×128	1
	Stability [21]	-	26×128	1
	Fluorescence [22]	-	26×128	1
Speech	LibriSpeech [17]	40	$1 \times 224 \times 224$	16×16
	Audio MNIST [1]	10	$1 \times 224 \times 224$	16×16
	Fluent Locations [13]	4	$1 \times 224 \times 224$	16×16
	Fluent Actions [13]	6	$1 \times 224 \times 224$	16×16
	Fluent Objects [13]	14	$1 \times 224 \times 224$	16×16
	Google Speech [28]	36	$1 \times 224 \times 224$	16×16
	VoxCeleb1 [15]	1251	$1 \times 224 \times 224$	16×16
RGB Image	ImageNet-32 [4]	1000	$3 \times 32 \times 32$	4×4
	CIFAR-10 [11]	10	$3 \times 32 \times 32$	4×4
	CUB [27]	200	$3 \times 32 \times 32$	4×4
	VGG Flowers [16]	102	$3 \times 32 \times 32$	4×4
	DTD [3]	47	$3 \times 32 \times 32$	4×4
	Traffic Sign [23]	43	$3 \times 32 \times 32$	4×4
	AirCraft [14]	102	$3 \times 32 \times 32$	4×4
	Wafer Map [29]	9	$3 \times 32 \times 32$	4×4

References

[1] S. Becker, M. Ackermann, S. Lapuschkin, K.-R. Müller, and W. Samek. Interpreting and explaining deep neural networks for classification of audio signals. *CoRR*, abs/1807.03418, 2018.

[2] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. The protein data bank. *Nucleic acids research*, 28(1):235–242, 2000.

[3] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi. Describing textures in the wild. In *CVPR*, pages 3606–3613, 2014.

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[5] S. El-Gebali, J. Mistry, A. Bateman, S. R. Eddy, A. Luciani, S. C. Potter, M. Qureshi, L. Richardson, G. A. Salazar, A. Smart, E. L. L. Sonnhammer, L. Hirsh, L. Paladin, D. Piovesan, S. C. E. Tosatto, and R. D. Finn. The pfam protein families database in 2019. *Nucleic Acids Research*, 2019.

[6] N. K. Fox, S. E. Brenner, and J.-M. Chandonia. Scope: Structural classification of proteins - extended, integrating scop and astral data and classification of new structures. *Nucleic Acids Research*, 2014.

[7] P. Helber, B. Bischke, A. Dengel, and D. Borth. Introducing eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. In *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*, 2018.

[8] P. Helber, B. Bischke, A. Dengel, and D. Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2019.

[9] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[10] M. S. Klausen, M. C. Jespersen, H. Nielsen, K. K. Jensen, V. I. Jurtz, C. K. Soenderby, M. O. A. Sommer, O. Winther, M. Nielsen, B. Petersen, et al. Netsurfp-2.0: Improved prediction of protein structural features by integrated deep learning. *Proteins: Structure, Function, and Bioinformatics*, 87(6):520–527, 2019.

[11] A. Krizhevsky et al. Learning multiple layers of features from tiny images, 2009.

[12] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

[13] L. Lugosch, M. Ravanelli, P. Ignoto, V. S. Tomar, and Y. Bengio. Speech model pre-training for end-to-end spoken language understanding. In G. Kubin and Z. Kacic, editors, *Proc. of Interspeech*, pages 814–818, 2019.

[14] S. Maji, E. Rahtu, J. Kannala, M. Blaschko, and A. Vedaldi. Fine-grained visual classification of aircraft. *Technical report*, 2013.

[15] A. Nagrani, J. S. Chung, and A. Zisserman. Voxceleb: a large-scale speaker identification dataset. In *INTERSPEECH*, 2017.

[16] M.-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729. IEEE, 2008.

[17] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur. Librispeech: an asr corpus based on public domain audio books. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2015.

[18] P. S. Pierre Baldi and D. Whiteson. Searching for exotic particles in high-energy physics with deep learning. *Nature Communications 5*, 2014.

[19] A. Reiss and D. Stricker. Introducing a new benchmarked dataset for activity monitoring. In *16th International Symposium on Wearable Computers, ISWC*, 2012.

[20] J. Ren, P. J. Liu, E. Fertig, J. Snoek, R. Poplin, M. A. DePristo, J. V. Dillon, and B. Lakshminarayanan. Likelihood ratios for out-of-distribution detection. In *Advances in Neural Information Processing Systems*, 2019.

[21] G. J. Rocklin, T. M. Chidyausiku, I. Goreshnik, A. Ford, S. Houliston, A. Lemak, L. Carter, R. Ravichandran, V. K. Mulligan, A. Chevalier, C. H. Arrowsmith, and D. Baker. Global analysis of protein folding using massively parallel design, synthesis, and testing. *Science*, 357 (6347):168–175, jul 2017.

[22] K. S. Sarkisyan, D. A. Bolotin, M. V. Meer, D. R. Usmanova, A. S. Mishin, G. V. Sharonov, D. N. Ivankov, N. G. Bozhanova, M. S. Baranov, O. Soylemez, et al. Local fitness landscape of the green fluorescent protein. *Nature*, 533(7603):397–401, 2016.

[23] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel. The german traffic sign recognition benchmark: a multi-class classification competition. In *The 2011 international joint conference on neural networks*, pages 1453–1460. IEEE, 2011.

[24] A. Tamkin, V. Liu, R. Lu, D. Fein, C. Schultz, and N. D. Goodman. Dabs: a domain-agnostic benchmark for self-supervised learning. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks*, 2021.

[25] A. Tamkin, G. Banerjee, M. Owda, V. Liu, S. Rammoorthy, and N. D. Goodman. Dabs 2.0: Improved datasets and algorithms for universal self-supervision. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS*, 2022.

[26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017.

[27] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset. *Technical Report CNS-TR-2010-001*, 2010.

[28] P. Warden. Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209*, 2018.

[29] M.-J. Wu, J.-S. R. Jang, and J.-L. Chen. Wafer map failure pattern recognition and similarity ranking for large-scale data sets. *IEEE Transactions on Semiconductor Manufacturing*, 28(1): 1–12, 2014.