BRIDGING GENERATIVE AND PREDICTIVE PARADIGMS VIA HIDDEN-SELF-DISTILLATION

Scott C. Lowe[1,*] **Anthony Fuller**[1,2], **Sageev Oore**[1,3], **Evan Shelhamer**[1,4], **Graham W. Taylor**[1,5]
[1]Vector Institute, [2]Carleton Uni., [3]Dalhousie Uni., [4]Uni. of British Columbia, [5]Uni. of Guelph

ABSTRACT

The landscape of self-supervised learning (SSL) is currently dominated by generative approaches (e.g., MAE) that reconstruct raw low-level data, and predictive approaches (e.g., I-JEPA) that predict high-level abstract embeddings. While generative methods provide strong grounding, they are computationally inefficient for high-redundancy modalities like vision, and their training objective does not prioritize learning high-level, conceptual features. Conversely, predictive methods often suffer from training instability due to their reliance on final-layer self-distillation. We introduce **Bootleg**, a method that bridges this divide by tasking the model with predicting continuous latent representations from *multiple hidden layers* of a teacher network. This hierarchical objective forces the model to capture features at varying levels of abstraction simultaneously. We demonstrate that Bootleg significantly outperforms comparable baselines on classification of ImageNet-1K and iNaturalist, and ADE20K segmentation (+10% over I-JEPA). This positions Bootleg as an ideal "representation interface" for next-generation multimodal models, where the optimal training signal lies neither at the pixel level nor the semantic peak, but in the rich intermediate hierarchy.

1 INTRODUCTION

The masked autoencoder (MAE) (He et al., 2022) and joint-embedding predictive architecture (I-JEPA) (Assran et al., 2023) are popular methods for image pretraining that rely on masking for self-supervised learning (SSL). Though these methods are not SOTA in image pretraining, their training paradigms are data agnostic, relying on masking instead of domain-specific data augmentations, enabling them to be adapted to many other modalities such as video (Wang et al., 2023; Bardes et al., 2024; Assran et al., 2025), audio (Huang et al., 2022; Fei et al., 2024; Tuncay et al., 2025; Yuksel et al., 2025), multimodal data (Bachmann et al., 2022; Chen et al., 2025; Lei et al., 2025), and other domains (Dong et al., 2024; Riou et al., 2024; Thimonier et al., 2025; Hachana & Rasheed, 2025).

In MAE, the model's training objective is to predict the *pixels* of the hidden patches (target $= x^{\text{mask}}$); whereas in I-JEPA, the models target the *final embeddings* of the hidden patches (target $= \bar{z}_B^{\text{mask}}$). I-JEPA's target embeddings are computed by processing the full image with a teacher-encoder model, which follows the (student) encoder by using an exponential moving average (EMA) of its weights. Conceptually, this choice of *target* is the key difference between MAE and I-JEPA.

However, both have their own advantages and disadvantages. MAE builds a representation learning model out of a generative modelling task, resulting in embeddings which are useful in pixel-space, but are less capable of representing high-level features. Consequently, MAE pretrained models require fine-tuning to be useful for downstream tasks. On the other hand, I-JEPA relies on *last layer* self-distillation, which is less stable to train because the targets are derived from the network we are training. This circularity means the targets are less stimulus-driven, which can result in the model being detached from reality and collapsing during training. In this work, we introduce *Bootleg*, which learns superior representations by reconstructing *multiple hidden layers* in its own teacher-encoder.

Illustrative experiments. As a proof-of-concept, we explore bridging the MAE and I-JEPA methods by simply sweeping the targets from the input layer (pixels) to the output layer (latents). Fig. 1 shows

*Correspondence: scott.lowe@vectorinstitute.ai

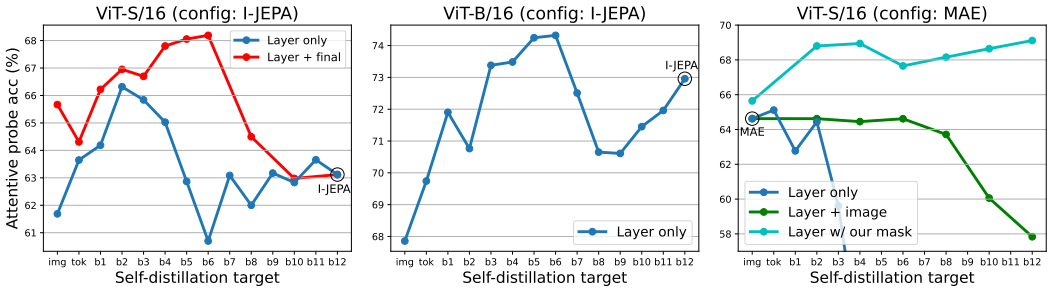

Figure 1: Bridging I-JEPA and MAE with targets across hidden layers. *Left:* We train ViT-S with I-JEPA, except for the target which we change to be a hidden layer (x-axis) of the teacher-encoder instead of its final output (blue curve) or in addition to the final output (red). We plot frozen attentive probe acc. on IN-1k (y-axis). *Middle:* Similar, but for ViT-B to verify at larger scale. *Right:* We train ViT-S using MAE, except we add EMA for self-distillation and change the target to be a hidden layer of the EMA model instead of the input pixels (blue) or in addition to the image pixels (green). We compare to a single target with our masking strategy (cyan). In all cases, self-distillation of a hidden target is able to improve on predicting only the input or the output.

the results of training ViT-S with a basic, i.e., ablated form of Bootleg: I-JEPA where the target is a layer of intermediate abstraction (a hidden layer within the teacher-encoder). We observe that it is superior to using the image or final embedding as the target, with the best self-distillation target being the output of the second layer of the teacher (left, blue line). A similar trend is seen for ViT-B (middle), where the best single target is the 6th layer. In both cases, the performance falls for deeper layers before rising again when the target is the output layer (12th layer). Furthermore, using multiple targets (a hidden layer of the teacher-encoder concatenated with its final layer), even with the same predictor capacity, yields superior performance to using a single hidden target (left, red line).

When we train with MAE but change the prediction target to a hidden layer, we find that performance decreases, with training instabilities causing a complete collapse in the model if the target is in the 4th layer or deeper (right, blue line). This issue is alleviated by predicting both the image pixels and a hidden layer (green line), which prevents the instabilities and model collapse. However, this is no better at the downstream task than predicting only pixels. When we change the masking strategy from uniform random to I-JEPA/Bootleg masks (cyan line), this sole change to the MAE is sufficient to stabilize training with hidden-self-distillation, and we observe a trend that is similar to the hidden-self-distillation version of I-JEPA (left).

2 METHOD

We now describe our hidden-self-distillation training method, Bootleg, illustrated in Fig. 2.

Objective. Given visible patches, the student-encoder creates embeddings that the predictor uses to forecast missing representations at masked locations. Our methodology is most similar to I-JEPA, with the main conceptual difference being the expansion of the targets to multiple hidden layers, instead of only the final layer.

Masking. We use block-structured masking (four rectangular regions), similar to I-JEPA, with improvements in implementation but no conceptual changes. The structured masks are essential: we find that uniform random masking (as in MAE) causes training collapse with self-distillation targets deeper than Block 2 (Appx. C.4).

Targets. The full input is processed by an EMA teacher-encoder. For each masked location j and each target depth $l \in L$, we collect the corresponding patch embedding $\bar{z}_{l,j}$ and independently z-score it: $t_{l,j} := \mathrm{zscore}(\bar{z}_{l,j})$. Targets across depths are concatenated into a single vector of length $|L| \times D$ per masked patch. We choose to use the output of every fourth block as a target for the student-predictor. Importantly, all targets are collected in a single forward pass through the teacher—no additional computation is required beyond indexing.

Student-Encoder. The student-encoder starts by embedding all unmasked patches into tokens. Our encoder has a further five global tokens: one class (CLS) token and four register tokens (Reg) (Darcet

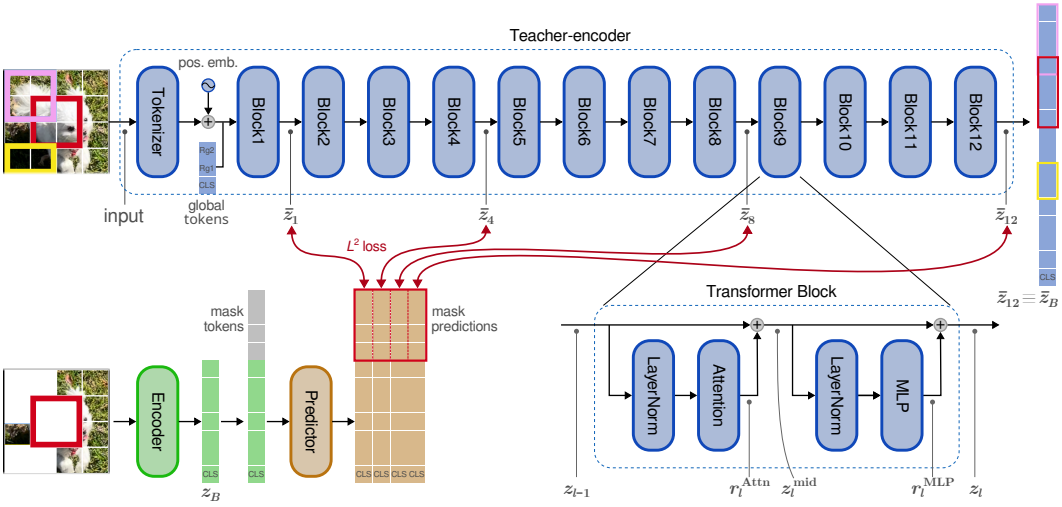

Figure 2: Multi-layer self-distillation with Bootleg. The teacher-encoder (blue), student-encoder (green), and predictor (orange) are ViTs, made of repeated transformer blocks. A schematic of a single transformer block is overlaid (bottom right). The teacher-encoder is an EMA of the student-encoder, and processes the full image. The student-encoder sees a subset of the image tokens and must create embeddings of them to facilitate the predictor. The predictor processes the embeddings to predict representations at multiple layers within the teacher-encoder.

et al., 2024); these are concatenated to the sequence after patch tokenization. From the output of the encoder, z_B, patch and CLS tokens are passed to the predictor; register tokens are discarded.

Predictor. We use a single predictor that maps encoder outputs plus mask tokens to the concatenated multi-layer target. Additionally, we prepend 4 predictor-register tokens—these serve a similar "global processing" role as the register tokens in the encoder, with their own learnable embeddings.

3 Experiments

To verify the utility of the Bootleg training procedures, we perform experiments with self-supervised training on ImageNet-1k (IN-1k) (Russakovsky et al., 2015). We pretrain all ViT models on 224×224 images with a patch size of 16×16. Full implementation details and hyperparameters are described in Appx. A. Evaluations are performed with frozen probes on IN-1k and iNaturalist-2021 (iNat21) classification (Van Horn et al., 2018), and frozen probes on ADE20K segmentation (Zhou et al., 2017). See Appx. B for methodological details.

Results. Our results, shown in Table 1, demonstrate the effectiveness of Bootleg for image classification. Across all model sizes, ViTs pretrained with Bootleg outperform the within-category competitors. This demonstrates the effectiveness of the pretraining methodology at learning to represent high-level image features. The largest margins ($>10\%$) were seen at the smallest model sizes. With full fine-tuning (Appx. Table 4) the models are better able to align their final embeddings with the classification task so there is less variation in performance among models, but Bootleg still performs best. The margin is larger if fine-tuning is done on a 1% subset of the data.

Bootleg performed best on all but one of the ADE20K segmentation probes. The frozen probes show Bootleg is much better able to perform pixel-level segmentation than its self-distillation competitor, I-JEPA, demonstrating that distillation of features along the visual pathway facilitates the model's ability to perform at a range of levels of granularity.

4 Intuitions, Discussion, and Future Work

Multi-scale grounding. By targeting early layers (which are stimulus-driven and spatially detailed) alongside deep layers (which are abstract and semantic), the model receives grounded training signal that prevents the degenerate feedback loops inherent in final-layer-only self-distillation. This is

Table 1: Results for masked SSL pretraining on IN-1k. We show the top-1 accuracy (%) for image classification on IN-1k and iNat21 using a probe on the frozen encoder (Patch, CLS, X-Blk). We also show mean-IoU (%) for semantic segmentation on ADE20K using a frozen encoder (kNN, Lin, Blk). We highlight the **best** and second best SSL method for each architecture size.

			IN-1k acc.			iNat21 acc.			ADE20K mIoU		
Arch	Method	Ep.	Patch	CLS	X-Blk	Patch	CLS	X-Blk	kNN	Lin	Blk
ViT-S/16	MAE	800	47.0	49.8	66.4	22.3	26.2	57.5	10.2	14.1	28.7
	CrossMAE	800	51.8	50.9	68.8	25.7	26.7	59.8	10.5	15.2	31.2
	data2vec 2.0	200	39.9	41.8	62.2	20.0	20.4	51.3	9.5	9.6	25.4
	I-JEPA	600	52.4	N/A	61.9	26.8	N/A	48.4	8.2	11.8	21.1
	Bootleg (ours)	600	**69.8**	**70.4**	**75.3**	**42.6**	**47.8**	**67.4**	**23.5**	**26.6**	**33.9**
ViT-B/16	MAE	1600	66.1	67.2	76.0	37.4	43.3	70.7	17.6	24.7	35.8
	CrossMAE	800	65.5	64.8	75.6	38.3	41.7	70.2	14.9	24.1	36.9
	data2vec 2.0	200	62.2	61.0	73.7	29.7	31.3	61.3	17.6	22.5	34.7
	I-JEPA	600	67.0	N/A	72.4	41.4	N/A	63.0	13.1	19.3	28.8
	Bootleg (ours)	600	**75.5**	**76.7**	**79.2**	**50.9**	**58.3**	**74.2**	**25.1**	**30.9**	**38.8**
ViT-L/16	MAE	1600	73.0	75.5	79.5	43.9	51.8	76.3	20.0	28.8	40.9
	CrossMAE	800	71.5	71.8	78.7	43.2	50.1	75.1	18.2	28.8	39.8
	data2vec 2.0	200	70.5	72.7	80.0	38.6	41.5	72.7	23.4	30.0	**43.0**
	I-JEPA	600	68.4	N/A	72.3	41.1	N/A	61.3	15.6	21.8	28.6
	Bootleg (ours)	600	**77.5**	**79.1**	**80.6**	**53.2**	**61.2**	**77.1**	**27.2**	**34.7**	41.2

analogous to predictive coding models of the brain (Rao & Ballard, 1999; Keller & Mrsic-Flogel, 2018), where neurons predict representations at all levels of the hierarchy.

Information bottleneck. With $|L|$ target layers, the predictor must reconstruct $|L| \times |M| \times D$ elements from a bottleneck of $|N| \times D_{\text{pred}}$ (with $|N| < |M|$), increasing the compression ratio and forcing the encoder to build richer, more structured representations (Tishby et al., 1999).

A modality-agnostic improvement. The core contribution of Bootleg—predicting multiple hidden-layer targets instead of the final layer only—is a change to the *training objective*, not the architecture or augmentation pipeline. This makes it directly applicable to any JEPA-family model. V-JEPA 2 (Assran et al., 2025) and audio JEPA models (Fei et al., 2024; Tuncay et al., 2025) all predict only final-layer features; multi-scale targets would provide fine-grained spatiotemporal or spectral detail (from early layers) alongside semantic content (from deep layers). Audio JEPA models currently lag behind reconstructive methods on fine-grained speech tasks; multi-layer targets could bridge this gap by including spectrally detailed early-layer representations. Furthermore, video data naturally decomposes into features with varying rates of change: high-frequency motion details and low-frequency semantic events. By forcing prediction across the depth of the network, Bootleg may naturally align with this temporal hierarchy: early-layer targets can capture rapid motion dynamics (high temporal frequency), whilst deep-layer targets capture slowly evolving semantic narratives (Menne et al., 2021; Beleza et al., 2023), potentially resolving the trade-off between motion-sensitivity and semantic-abstraction in video foundation models (Sun et al., 2025).

Related works. Data2vec (Baevski et al., 2022; 2023) showed multi-layer teacher targets improve over final-layer-only across speech, vision, and language, but uses averaging over the final layers for the target rather than prediction of multiple, distinct, embeddings. We find predicting individual layers spread across the whole visual hierarchy is superior to averaging the final layers (see Appx. C.1.1). MC-JEPA (Bardes et al., 2023) similarly showed that predicting diverse feature types in a shared encoder is beneficial. Our approach preserves the full hierarchy via independent per-layer prediction.

Limitations. Our current experiments are limited to static images. Extending to video (spatiotemporal masking), audio (spectrogram masking), and multimodal settings is a natural next step and the subject of ongoing work. The masking strategy remains critical: block-structured masks are necessary to prevent training collapse with self-distillation, and optimal masking may differ across domains.

ACKNOWLEDGEMENTS

Resources used in preparing this research were provided, in part, by the Province of Ontario, the Government of Canada through CIFAR, and companies sponsoring the Vector Institute.

REFERENCES

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)*, pp. 15619–15629, 2023. doi:10.1109/CVPR52729.2023.01499.

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. doi:10.48550/arxiv.2506.09985.

Roman Bachmann, David Mizrahi, Andrei Atanov, and Amir Zamir. MultiMAE: Multi-modal multi-task masked autoencoders. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner (eds.), *Computer Vision – ECCV 2022*, pp. 348–367, Cham, 2022. Springer Nature Switzerland. ISBN 978-3-031-19836-6. doi:10.1007/978-3-031-19836-6_20. URL https://www.ecva.net/papers/eccv_2022/papers_ECCV/html/7102_ECCV_2022_paper.php.

Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. data2vec: A general framework for self-supervised learning in speech, vision and language. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 1298–1312. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/baevski22a.html.

Alexei Baevski, Arun Babu, Wei-Ning Hsu, and Michael Auli. Efficient self-supervised learning with contextualized target representations for vision, speech and language. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 1416–1429. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/baevski23a.html.

Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. BEiT: BERT pre-training of image transformers. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=p-BhZSz59o4.

Adrien Bardes, Jean Ponce, and Yann LeCun. MC-JEPA: A joint-embedding predictive architecture for self-supervised learning of motion and content features. *arXiv preprint arXiv:2307.12698*, 2023. doi:10.48550/arxiv.2307.12698.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. doi:10.48550/arxiv.2404.08471. Featured Certification.

Suzana Rita Alves Beleza, Erica K. Shimomoto, Lincon S. Souza, and Kazuhiro Fukui. Slow feature subspace: A video representation based on slow feature analysis for action recognition. *Machine Learning with Applications*, 14:100493, 2023. ISSN 2666-8270. doi:10.1016/j.mlwa.2023.100493.

Delong Chen, Mustafa Shukor, Theo Moutakanni, Willy Chung, Jade Yu, Tejaswi Kasarla, Yejin Bang, Allen Bolourchi, Yann LeCun, and Pascale Fung. VL-JEPA: Joint embedding predictive architecture for vision-language. *arXiv preprint arXiv:2512.10942*, 2025. doi:10.48550/arxiv.2512.10942.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1597–1607. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/chen20j.html.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=2dnO3LLiJ1.

Timothée Darcet, Federico Baldassarre, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Cluster and predict latents patches for improved masked image modeling. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=Ycmz7qJxUQ.

Zijian Dong, Ruilin Li, Yilei Wu, Thuan Tinh Nguyen, Joanna Su Xian Chong, Fang Ji, Nathanael Ren Jie Tong, Christopher Li Hsian Chen, and Juan Helen Zhou. Brain-JEPA: Brain dynamics foundation model with gradient positioning and spatiotemporal masking. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 86048–86073. Curran Associates, Inc., 2024. doi:10.52202/079017-2732. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/9c3828adf1500f5de3c56f6550dfe43c-Paper-Conference.pdf.

Zhengcong Fei, Mingyuan Fan, and Junshi Huang. A-JEPA: Joint-embedding predictive architecture can listen. *arXiv preprint arXiv:2311.15830*, 2024. doi:10.48550/arxiv.2311.15830.

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. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=JT2KMuo2BV.

Rafik Hachana and Bader Rasheed. Using a joint-embedding predictive architecture for symbolic music understanding. In *AI for Music Workshop*, 2025. URL https://openreview.net/forum?id=lieErtGZb6.

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)*, pp. 15979–15988, 2022. doi:10.1109/CVPR52688.2022.01553.

Carlos Hinojosa, Shuming Liu, and Bernard Ghanem. ColorMAE: Exploring data-independent masking strategies in masked autoencoders. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (eds.), *Computer Vision – ECCV 2024*, pp. 432–449, Cham, 2025. Springer Nature Switzerland. ISBN 978-3-031-72661-3. doi:10.1007/978-3-031-72661-3_25. URL https://www.ecva.net/papers/eccv_2024/papers_ECCV/html/3072_ECCV_2024_paper.php.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Po-Yao Huang, Hu Xu, Juncheng Li, Alexei Baevski, Michael Auli, Wojciech Galuba, Florian Metze, and Christoph Feichtenhofer. Masked autoencoders that listen. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 28708–28720. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/b89d5e209990b19e33b418e14f323998-Paper-Conference.pdf.

Georg B. Keller and Thomas D. Mrsic-Flogel. Predictive processing: A canonical cortical computation. *Neuron*, 100(2):424–435, Oct 2018. ISSN 0896-6273. doi:10.1016/j.neuron.2018.10.003.

Hongyang Lei, Xiaolong Cheng, Qi Qin, Dan Wang, Huazhen Huang, Qingqing Gu, Yetao Wu, and Luo Ji. M3-JEPA: Multimodal alignment via multi-gate MoE based on the joint-embedding predictive architecture. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*. PMLR, 2025. URL https://proceedings.mlr.press/v267/lei25b.html.

Max Menne, Merlin Schüler, and Laurenz Wiskott. Exploring slow feature analysis for extracting generative latent factors. In *Proceedings of the 10th International Conference on Pattern Recognition Applications and Methods - ICPRAM*, pp. 120–131. INSTICC, SciTePress, 2021. ISBN 978-989-758-486-2. doi:10.5220/0010391401200131.

Rajesh P. N. Rao and Dana H. Ballard. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature Neuroscience*, 2(1):79–87, Jan 1999. ISSN 1546-1726. doi:10.1038/4580.

Alain Riou, Stefan Lattner, Gaëtan Hadjeres, Michael Anslow, and Geoffroy Peeters. Stem-JEPA: A Joint-Embedding Predictive Architecture for Musical Stem Compatibility Estimation. In *Proceedings of the 25th International Society for Music Information Retrieval Conference*, San Francisco, nov 2024. ISMIR.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, Apr 2015. doi:10.1007/s11263-015-0816-y.

Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7242–7252, 2021. doi:10.1109/ICCV48922.2021.00717.

Shengkai Sun, Zefan Zhang, Jianfeng Dong, Zhiyong Cheng, Xiaojun Chang, and Meng Wang. Towards efficient general feature prediction in masked skeleton modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12212–12221, October 2025. URL https://openaccess.thecvf.com/content/ICCV2025/papers/Sun_Towards_Efficient_General_Feature_Prediction_in_Masked_Skeleton_Modeling_ICCV_2025_paper.pdf.

Hugo Thimonier, José Lucas De Melo Costa, Fabrice Popineau, Arpad Rimmel, and Bich-Liên Doan. T-JEPA: Augmentation-free self-supervised learning for tabular data. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=gx3LMRB15C.

Naftali Tishby, Fernando C. Pereira, and William Bialek. The information bottleneck method. In *Proceedings of the 37th Allerton Conference on Communication, Control, and Computing*, pp. 368–377, 1999. doi:10.48550/arXiv.physics/0004057.

Ludovic Tuncay, Etienne Labbé, Emmanouil Benetos, and Thomas Pellegrini. Audio-JEPA: Joint-embedding predictive architecture for audio representation learning. *arXiv preprint arXiv:2507.02915*, 2025. doi:10.48550/arxiv.2507.02915.

Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The iNaturalist species classification and detection dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8769–8778, Los Alamitos, CA, USA, June 2018. IEEE Computer Society. doi:10.1109/CVPR.2018.00914. URL https://openaccess.thecvf.com/content_cvpr_2018/papers/Van_Horn_The_INaturalist_Species_CVPR_2018_paper.pdf.

Shashanka Venkataramanan, Valentinos Pariza, Mohammadreza Salehi, Lukas Knobel, Spyros Gidaris, Elias Ramzi, Andrei Bursuc, and Yuki M. Asano. Franca: Nested matryoshka clustering for scalable visual representation learning. *arXiv preprint arXiv:2507.14137*, 2025. doi:10.48550/arxiv.2507.14137.

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)*, pp. 14549–14560, 2023. doi:10.1109/CVPR52729.2023.01398.

Goksenin Yuksel, Pierre Guetschel, Michael Tangermann, Marcel van Gerven, and Kiki van der Heijden. WavJEPA: Semantic learning unlocks robust audio foundation models for raw waveforms. *arXiv preprint arXiv:2509.23238*, 2025. doi:10.48550/arxiv.2509.23238.

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ADE20K dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5122–5130, 2017. doi:10.1109/CVPR.2017.544.

APPENDICES

A PRETRAINING HYPERPARAMETERS

We selected the hyperparameters for pretraining Bootleg primarily by referring to those used in MAE (He et al., 2022) and I-JEPA (Assran et al., 2023). The hyperparameters used for Bootleg are shown in Table 2, with MAE and I-JEPA listed for comparison.

In our preliminary experiments (c.f. Fig. 1), we found training using the MAE hyperparameter configuration with the I-JEPA masking strategy and a hidden target for self-distillation outperformed similar experiments using the I-JEPA hyperparameter configuration. We thus initialized our hyperparameter configuration with that of MAE. Through a series of preliminary experiments conducted with ViT-S, we considered changing hyperparameters to that of I-JEPA or V-JEPA (Bardes et al., 2024). We additionally considered adding register tokens (Darcet et al., 2024) since this had been shown to be highly beneficial for networks pretrained with DINO (Darcet et al., 2024). We found adding (separate) registers to each of the encoder and predictor networks was beneficial.

We then conducted a series of line searches with ViT-S, 300 ep., on the predictor size, learning rate schedule, weight decay, and number of warm-up steps. To assess the scaling of the predictor size relative to the encoder size, we conducted a small number of experiments with ViT-B and -L encoders, varying both the width and depth of the encoder. To assess the learning rate, which may vary depending on model size, we conducted limited experiments on ViT-B and -L using the schedule established with ViT-S. Our final hyperparameters are shown in Table 2. We found that a larger crop size, longer LR warmup duration, and higher maximum and minimum LR values improved performance of the resulting model. We scale the warmup duration based on the total number of epochs, E, as $E_{\text{wu}} = 33 + 0.12\,E$.

We changed the model seed used between each round of our hparam search to avoid overfitting to a specific model. The final evaluations shown in the results tables were performed on models trained from scratch using a fresh seed which had not been included in the hyperparameter search.

Some hyperparameters vary between model sizes. The resulting values are shown in Table 3. We use a ViT predictor of depth 10 with width equal to half that of the encoder, with heads equal to same as the encoder with a minimum of 16 heads. We found ViT-S needed a higher initial and final LR ($1/30$ of the maximum) than ViT-B and -L ($1/100$ of the maximum LR).

Our results include ViT-S models, which is outside the original training recipes of MAE and I-JEPA, hence we additionally share the configs we used for these in Table 3. For MAE, we follow the ViT-S recipe of (Fu et al., 2025). For I-JEPA, we halve the width of the predictor to maintain the same capacity relative to the encoder as used for ViT-B, and let the number of heads equal that of the encoder following the rule from I-JEPA (Assran et al., 2023).

B EVALUATION DETAILS

B.1 IMAGENET-1K CLASSIFICATION

After SSL pretraining of the ViT model with Bootleg on IN-1k (Russakovsky et al., 2015), we evaluate the model's performance at supervised IN-1k and iNat21 classification.

B.1.1 FROZEN PROBES FOR CLASSIFICATION

Method We investigate the alignment of the unmodified embeddings from the pretrained encoder with the image classification task. This is done with a probe of the frozen encoder, which we conduct in three ways. Firstly, we consider the long-standing linear probe of the average patch embeddings (column "**Patch**" in tables). We discard the CLS token and register tokens, take the average of the patch embeddings, apply batch norm, and a single learnable linear layer to predict the IN-1k classes. Secondly, we consider the performance of a linear probe of the CLS token embedding (column "**CLS**" in tables). We discard the patch tokens and register tokens, apply batch norm, and train a single learnable linear layer on the CLS token embedding. Thirdly, we consider the performance of a non-linear attentive probe (column "**X-Blk**" in tables). This was performed following the

Table 2: Hyperparameter configuration comparison between Bootleg, I-JEPA, MAE for ViT-B/16. Learning rate (LR) values are tabulated for a batch size of 2048 samples, and linearly scaled to the batch size used for training. Mask seen rate and target rate are empirically estimated (mean $\pm$ stdev) for I-JEPA and Bootleg with a batch size per GPU of 256.

	Hyperparameter	MAE	I-JEPA	Bootleg (ours)
Enc.	Architecture	ViT-B/16	ViT-B/16	ViT-B/16
	Depth	12	12	12
	Width	768	768	768
	Attention heads	12	12	12
	Patch size	16	16	16
	CLS tokens	1	0	1
	Register tokens	0	0	4
Pred.	Depth	8	6	10
	Width	512	384	384
	Attention heads	16	12	16
	Register tokens	0	0	4
Data	Crop size	RandCrop(0.2, 1.0)	RandCrop(0.3, 1.0)	RandCrop(0.35, 1.0)
	Horizontal flip	✓	✗	✓
	Interpolation	Bicubic	Bilinear	Bicubic
	Input size	224×224	224×224	224×224
Mask	Strategy	Uniform random	4 rectangular blocks	4 rectangular blocks
	Mask seen rate	0.25	0.243 ± 0.037	0.292 ± 0.069
	Mask target rate	0.75	0.445 ± 0.070	0.475 ± 0.076
Train	Target(s)	Image pixels	Final (block 12) output	Block 1, 4, 8, 12 outputs
	Target standardization	Z-score	Z-score	Z-score
	Target length	768	768	3072
	Loss	Mean squared error	Smooth L1	Mean squared error
	Optimizer	AdamW($\beta = (0.9, 0.95)$)	AdamW($\beta = (0.9, 0.999)$)	AdamW($\beta = (0.9, 0.95)$)
	EMA initial	N/A	0.996	0.9985
	EMA final	N/A	1.0	0.9985
	LR schedule	Cosine annealing	Cosine annealing	Cosine annealing
	LR schedule warmup	40 epochs	40 epochs	105 epochs
	LR initial	0.0	0.0002	0.00003
	LR maximum	0.0012	0.001	0.003
	LR final	0.0	0.000001	0.00003
	WD initial	0.05	0.04	0.05
	WD final	0.05	0.40	0.05
	Schedule stretch	1.0	1.0	1.0
	Batch size	4096	2048	2048
	Num epochs	1600	600	600

implementation in V-JEPA (Bardes et al., 2024; Assran et al., 2025): we keep the patch, CLS, and any register embeddings produced by the model; these are passed to a cross-attention block with a single learnable query vector, and an MLP sub-block, followed by a linear head. We use the same width and number of attention heads as in the pretrained ViT. All probes were trained with cross-entropy for 20 epochs, batch size 1024, and a simultaneous hyperparameter sweep of 25 learning rate and weight decay configurations. For the linear probes, we use a random crop over scales $[0.3, 1.0]$ (random aspect ratio), and basic augmentations (horizontal flip, colour jitter) only. As in previous work (He et al., 2022), we discard the final norm from the encoder when taking the embeddings from the pretrained model.

B.1.2 Fine-tuning for classification

We fine-tuned the pretrained models on IN-1k using either the full labels, or a 1% subset of the labels as described below.

Method. Our methodology for full-fine tuning on IN-1k follows that recommended in the MAE GitHub repository and used by CrossMAE (Fu et al., 2025).

We performed low-shot fine-tuning on 1% of IN-1k using the subset proposed by SimCLR (Chen et al., 2020) and the methodology from I-JEPA (Assran et al., 2023). We adapted the methodology to sweep over five maximum learning rates, $\eta_{\text{max}} \in [5\times10^{-6}, 1.5\times10^{-5}, 5\times10^{-5}, 1.5\times10^{-4}, 5\times10^{-4}]$, and report the best performance over these learning rates.

Table 3: Hyperparameter changes across encoder sizes ViT-S, -B, -L for MAE, I-JEPA, and Bootleg models. Learning rate (LR) values are tabulated for a batch size of 2048 samples, and linearly scaled to the batch size used for training.

		MAE			I-JEPA			Bootleg (ours)		
	Hyperparameter	ViT-S	ViT-B	ViT-L	ViT-S	ViT-B	ViT-L	ViT-S	ViT-B	ViT-L
Enc.	Depth	12	12	24	12	12	24	12	12	24
	Width	384	768	1024	384	768	1024	384	768	1024
	Attention heads	6	12	16	6	12	16	6	12	16
	Patch size	16	16	16	16	16	16	16	16	16
	CLS tokens	1	1	1	0	0	0	1	1	1
	Register tokens	0	0	0	0	0	0	4	4	4
Pred.	Depth	8	8	8	6	6	12	10	10	10
	Width	256	512	512	192	384	384	192	384	512
	Attention heads	8	16	16	6	12	16	16	16	16
	Register tokens	0	0	0	0	0	0	4	4	4
Train	Target(s)	Input	Input	Input	Final	Final	Final	b1,4,8,12	b1,4,8,12	b1,4,8,12,16,20,24
	Target length	384	768	1024	384	768	1024	1536	3072	7168
	LR initial	0.0	0.0	0.0	2e-4	2e-4	2e-4	1e-4	3e-5	3e-5
	LR maximum	1.2e-3	1.2e-3	1.2e-3	1e-3	1e-3	1e-3	3e-3	3e-3	3e-3
	LR final	0.0	0.0	0.0	1e-6	1e-6	1e-6	1e-4	3e-5	3e-5

We additionally evaluated the models at low-rank (parameter efficient) fine-tuning. For this we use the LoRA methodology (Hu et al., 2022) with rank $= 8$ and $\alpha = 16$. We fine-tune the attention QKV weights, attention projector, and MLP weights from every block on the 1% low-shot subset of IN-1k. Learning rates were swept over $\eta_{\text{max}} \in [1.5{\times}10^{-5}, 5{\times}10^{-5}, 1.5{\times}10^{-4}, 5{\times}10^{-4}, 1.5{\times}10^{-3}]$, and we report the best performance over these learning rates.

All fine-tuned models were evaluated on IN-1k validation set, with an 87.5% centre crop (resize to 256×256, then crop to 224×224).

Results. As shown in Table 4, Bootleg achieves the best or near-best fine-tuning performance across all settings. On IN-1k 100% fine-tuning, Bootleg outperforms all baselines at every model size, with particularly large margins at ViT-S (+0.9 over the next best) and ViT-L (+0.7). The advantage is most pronounced in the low-data regime: on 1% fine-tuning, Bootleg surpasses the next-best method by +8.6 (ViT-B) and +12.5 (ViT-S) points, demonstrating that hidden-self-distillation produces representations that are especially effective when labelled data is scarce. LoRA fine-tuning follows a similar pattern, with Bootleg matching or exceeding full fine-tuning of I-JEPA despite updating far fewer parameters.

Data2vec 2.0 full fine-tuning was sometimes unstable at the learning rates used for these experiments (collapsed model attaining 0.1% acc.) and would benefit from a reduced learning rate. As described above, we used the methodology and learning rate given by existing literature (Fu et al., 2025). Unfortunately, we could not sweep the learning rate due to the computational demands of full-fine tuning.

B.2 ADE20K SEGMENTATION

To check the performance of ViTs pretrained with Bootleg on a standard dense-prediction task, we run experiments on the ADE20K semantic segmentation benchmark (Zhou et al., 2017).

B.2.1 FROZEN PROBES FOR SEGMENTATION

Method—kNN We probe using the kNN methodology from CAPI (Darcet et al., 2025). For this probe, images from the training partition are encoded without augmentations. The embeddings and pixel-wise segmentation labels for each patch token form the kNN training set. Inference is performed by finding the k nearest neighbours for each patch in the image, and then for each pixel in the patch predicting the most common label across that pixel location in the k neighbours. We sweep 4 neighbourhood sizes $k \in [1, 3, 10, 30]$, using both cosine and L2 distance metrics.

Method—Lin and Blk We probe using a Segmenter (Strudel et al., 2021) head trained atop a frozen backbone. We consider two heads, trained separately. "**Lin**", a linear projector from the patch embeddings to the Segmenter; or "**Blk**", with one randomly initialized transformer block before the

Table 4: Fine-tuning results on IN-1k classification (top-1 accuracy, %) and ADE20K semantic segmentation (mIoU, %). IN-1k results include full fine-tuning on 100% and 1% of training data, and LoRA fine-tuning on 1%. Ep: number of pretraining epochs.

			IN-1k acc.			ADE20K mIoU
			Full-FT		LoRA	Full-FT
Arch	Method	Ep.	100%	1%	1%	100%
ViT-S/16	MAE	800	78.2	37.1	33.9	39.2
	CrossMAE	800	79.8	40.4	37.5	<u>42.2</u>
	data2vec 2.0	200	79.9	39.3	36.8	40.5
	I-JEPA	600	<u>79.9</u>	<u>48.6</u>	<u>47.3</u>	36.1
	Bootleg (ours)	600	**80.8**	**61.1**	**60.4**	**44.3**
ViT-B/16	MAE	1600	82.7	55.9	53.5	44.0
	CrossMAE	800	82.9	52.3	49.6	<u>45.0</u>
	data2vec 2.0	200	<u>83.3</u>	58.9	58.1	0.1
	I-JEPA	600	82.7	<u>61.9</u>	<u>62.2</u>	29.8
	Bootleg (ours)	600	**83.9**	**70.5**	**69.9**	**46.6**
ViT-L/16	MAE	1600	<u>84.7</u>	67.7	67.5	<u>50.7</u>
	CrossMAE	800	84.3	62.7	61.7	49.6
	data2vec 2.0	200	0.1	<u>73.2</u>	**74.1**	**51.8**
	I-JEPA	600	82.0	63.8	63.8	31.9
	Bootleg (ours)	600	**85.4**	**73.4**	<u>73.2</u>	48.3

projector. Our training strategy is based on the BEiT evaluation methodology (Bao et al., 2022): we train for 128 epochs with a batch size of 64, and augment the data with random resized cropping, horizontal flips, cut-out, and colour jitter. For the linear projector, we halve the magnitude of the colour jitter.

We sweep 4 learning rates for each head type, and report the best performance. For Lin probes, we use $\eta_{\text{max}} \in [2.5\times10^{-4}, 7.5\times10^{-4}, 2.5\times10^{-3}, 7.5\times10^{-3}]$; for Blk we use $\eta_{\text{max}} \in [2.5\times10^{-5}, 7.5\times10^{-5}, 2.5\times10^{-4}, 7.5\times10^{-4}]$. We empirically found these LR ranges were in the optimal range for the models considered.

All frozen evaluation was performed using images at 224×224 resolution. As the pretrained model uses sinusoidal position embeddings at 224×224 resolution (14×14 patches), it lacks the flexibility to correctly process other image sizes without retraining.

Results. As shown in Table 1, Bootleg performs competitively at semantic segmentation. The frozen probes show Bootleg has much better able to perform pixel-level segmentation than its self-distillation competitor, I-JEPA, demonstrating that distillation of features along the visual pathway facilitates the model's ability to perform at a range of levels of granularity.

B.2.2 Fine-tuning for segmentation

Method. For ADE20K fine-tuning, we add a single-block MaskTransformer decoder (Strudel et al., 2021) to the encoder. Both the encoder and decoder are fine-tuned end-to-end for 128 epochs with a batch size of 32, using the AdamW optimizer. The hyperparameters used, including learning rate, were selected on a prototyping model and not tuned for our final model(s) on which we report the final results.

Fine-tuning evaluation was performed using images at 512×512 resolution. We interpolate the position embeddings to accommodate the processing of 512×512 px images, a change which the model is quickly able to adapt to during fine-tuning.

For evaluation, images are simply resized to 512×512, with no other transformations applied.

Results. As shown in Table 4, we find on ADE20K fine-tuning, Bootleg leads at ViT-S and ViT-B; at ViT-L, data2vec 2.0 and MAE are stronger.

C Ablations

We ablate key elements of our method, compare extensions of other methods, closely study the differences with I-JEPA.

C.1 Choice of target layers

The core mechanism of Bootleg is the self-distillation of hidden layers—which poses the question of which hidden layers to distill. In preliminary experiments, we explored the performance of ViT-S, -B, -L with different numbers of hidden targets for self-distillation during pretraining. In addition to predicting the output of various hidden blocks, z_l, we explored the performance when predicting the mid-block representations, z_l^{mid}, and the residual terms r_l^{Attn} and r_l^{MLP}. For ViT-S, we find two peaks in performance: one when using 3–4 equispaced block outputs as targets, and another when targeting multiple layers within every block. However, targeting the residual layers was only sometimes beneficial, and only seen for ViT-S. Overall, we found targeting the output of every fourth block in the teacher is a rule-of-thumb which gives consistently strong performance, and often the best performance of the configurations tried.

C.1.1 Spacing out targets

We considered whether to choose targets which are grouped together in a contiguous group of early blocks (the outputs of the first third of the transformer blocks), mid blocks (the second third), or late blocks (the final third). As shown in Table 5, using equispaced targets from across a broader range of depths within the transformer, corresponding to a range of levels of abstraction in representation, is superior to using either early, mid, or late blocks exclusively.

Table 5: Target construction strategy. We compare four spaced-out targets (ours) to consecutive targets from the early, middle, or final layers of the network. We also consider two options of how to merge targets: either standardize and concatenate (ours) or standardize, average, and re-standardize (data2vec-style). Spaced-out, concatenated, targets yield the best training targets. Experiments trained for 300 ep. on IN-1k and evaluated with frozen probe.

		IN-1k acc.			ADE20K mIoU		
Targets	Merge op.	Patch	CLS	X-Blk	kNN	Lin	Blk
Pixel targets (MAE-style)	N/A	52.9	51.0	66.7	13.6	17.7	27.5
Blocks 1–4 (early only)	Concat	58.0	57.1	69.5	16.5	21.0	30.1
Blocks 5–8 (mid only)	Concat	61.6	61.7	72.2	18.6	23.0	33.0
Blocks 3–12 (last 10)	Concat	64.6	66.0	73.4	20.9	24.5	33.4
Blocks 9–12 (late only)	Concat	60.2	60.8	69.7	16.0	19.7	29.3
Blocks $\{1, 4, 8, 12\}$ (spaced)	Concat	**67.8**	**68.9**	**74.4**	**22.1**	**26.6**	**34.2**
Blocks 3–12 (last 10; d2v-style)	Average	44.1	37.9	59.9	8.8	12.8	24.2
Blocks $\{1, 4, 8, 12\}$ (spaced)	Average	57.2	54.6	69.8	15.6	19.8	30.5

C.1.2 Concatenated versus averaged hidden embeddings

We compared the effect of our multi-target method to that of data2vec 2.0 (Baevski et al., 2023) as follows. We ablated the Bootleg ViT-S training recipe by changing our target concatenation step to instead apply instance norm to the embeddings an then take the average, as per data2vec 2.0. We considered either keeping the same set of targets as in Bootleg, or changing them to the set of hidden layers averaged in data2vec 2.0. As shown in Table 5, we found our training target construction method was superior to that of data2vec 2.0.

C.2 Bootleg editions of other Masked Image Modelling methods

We explore the effect of adding our main conceptual contribution—self-distillation of hidden layers—to other masked image modelling methods with minimal other changes (Table 6). We train a ViT-S for 300 epochs on IN-1k using either the MAE, CrossMAE, or I-JEPA framework, then evaluate with frozen probes on IN-1k, as described in Appx. B.1.1.

Adding EMA. To facilitate self-distillation, we need to add an EMA teacher to MAE and CrossMAE; we first evaluate the performance of the EMA encoder without using it for distillation. Its performance is negligibly different from that of the encoder without EMA.

Masking. As we show in Sec. 1, the masking strategy of MAE is a limiting factor for the adoption of self-distillation of targets deeper than Block 2. We thus change its masking strategy to our implementation of I-JEPA masking, which improved performance. For MAE, we also apply the predictor once per mask region, allowing self-attention interactions within the tokens of each region. For CrossMAE, the use of cross- instead of self-attention means there is no interaction between the mask tokens, and so this is not a consideration; we use the concatenation of the mask tokens from all four regions for efficiency[1].

Hidden-self-distillation. With these changes in place, we next change the target from the image pixels (MAE and CrossMAE) and final layer (I-JEPA) to our hidden distillation targets. This greatly increases the performances for MAE and CrossMAE ($+10\%$ Patch and CLS probes, $+6\%$ to X-Blk probes) and I-JEPA ($+6\%$). These findings indicate masked image modelling can in general benefit from using our hidden-self-distillation, provided the masking strategy supports it.

Table 6: Effect of combining key Bootleg components with existing pretraining methods. Experiments conducted with ViT-S, 300 ep., and evaluated with frozen probe on IN-1k (top-1 acc, %). Modifications (and deltas) are cumulative. See Appx. C.2 for details.

		IN-1k acc.			ADE20K mIoU	
Base	Modification	Patch	CLS	X-Blk	kNN	Lin
MAE	Unmodified	44.7	47.0	65.1	9.3	13.1
	+EMA	44.8 +0.1	46.9 −0.1	65.1 −0.0	9.1 −0.2	13.1 −0.0
	+our masking	51.2 +6.5	50.7 +3.8	66.0 +0.9	12.2 +3.1	16.6 +3.5
	+change targets	62.4 +11.2	62.9 +12.2	72.1 +6.1	19.5 +7.3	24.1 +7.4
CrossMAE	Unmodified	45.7	44.4	65.2	8.5	12.7
	+EMA	45.7 +0.0	44.3 −0.1	65.3 +0.1	8.6 +0.1	12.7 +0.0
	+our masking	51.5 +5.7	52.5 +8.2	66.6 +1.3	10.3 +1.7	15.8 +3.1
	+change targets	60.8 +9.4	61.6 +9.1	71.8 +5.3	14.6 +4.4	21.4 +5.5
I-JEPA	Unmodified	53.4	N/A	62.3	10.0	13.5
	+change targets	59.5 +6.1	N/A	69.3 +7.0	15.1 +5.2	19.3 +5.7

C.3 Interpolating between I-JEPA and Bootleg

We carefully identify and evaluate the differences between Bootleg and the closest existing masked image modelling method: I-JEPA. As shown in Table 7, we find changing the targets to be the outputs of multiple blocks is the largest single improvement made to the I-JEPA training configuration, followed by improving the implementation of the masking strategy and the addition of a CLS token. However, our final Bootleg training recipe is more heavily impacted by ablating features such as the larger predictor (which facilitates processing additional targets) than ablating the number of targets.

C.4 Changing masking strategy

We experimented with ablating our masking strategy entirely, using methods from recent papers. The results are shown in Table 8.

When training with uniform random masks as per MAE, training became unstable after 160 epochs and the model collapsed. Using "green" structured noise from ColorMAE (Hinojosa et al., 2025) or inverse-blocks as per data2vec (Baevski et al., 2023), training was unstable and performance suffered as a result, but not as badly as when using random noise. In contrast to this, cyclic block masking, used by CAPI and Franca (Darcet et al., 2025; Venkataramanan et al., 2025), yielded stable training and even out-performed the I-JEPA implementation of multi-block masking.

[1]Note that the overlap of mask regions means there are redundant mask tokens for the same patch location for the CrossMAE predictor; these serve only to increase the loss weighting on resampled mask locations, and for consistency we did not deduplicate these.

Table 7: Ablation of the differences between I-JEPA and Bootleg. Each row isolates a single change to the training configuration. We show the performance of a model trained with a config equal to I-JEPA plus one change from Bootleg ("Only with"), and with a config equivalent to changing everything from the I-JEPA to Bootleg config except for one component ("Only without"). Next to each of these is the change in probe accuracy observed when adding that first component to I-JEPA's config, and when adding it as the final component to reach Bootleg's complete config. Changes between rows are not cumulatively stacked on top of each other. Experiments performed with ViT-S, 300 ep. See Tables 2 and 3 for the hyperparameter values compared.

	IN-1k acc., X-Blk		ADE mIoU, Lin		
Ablation	Only with	Only without	Only with	Only without	Avg Δ
I-JEPA (= with none)	62.3		13.5		
Data transforms: +hflip, ↑min crop size	63.2 +0.8	74.2 +0.2	13.3 −0.2	25.8 +0.8	+0.4
Add CLS token	65.9 +3.5	74.1 +0.3	17.4 +3.9	25.6 +1.0	+2.2
Add 4/4 reg tokens to encoder/predictor	64.7 +2.4	73.9 +0.5	15.5 +1.9	25.6 +1.0	+1.4
Predictor size: ↑ depth, ↑ heads	64.1 +1.8	72.5 +1.9	13.9 +0.3	23.6 +3.0	+1.8
Masking: implementation improvements	68.2 +5.9	72.6 +1.8	18.0 +4.5	24.5 +2.2	+3.6
Targets: $T = \{12\} \rightarrow \{1, 4, 8, 12\}$	69.3 +7.0	73.9 +0.4	19.3 +5.7	25.2 +1.4	+3.6
Loss: Smooth L1 → L2	65.1 +2.7	74.5 −0.1	14.9 +1.3	25.5 +1.1	+1.3
Hyperparams: ↑ LR, ↑ WU, ↓ WD	63.7 +1.3	73.1 +1.2	13.0 −0.5	24.1 +2.5	+1.2
Bootleg (= with all)		74.4		26.6	

The key difference between these masking strategies is the size of the contiguous blocks which are masked out—larger mask sizes yielded better performance. We hypothesize this is because neighbouring patch tokens have highly correlated activations, especially deeper in the visual hierarchy. Hence it is important that the majority of target patches are not adjacent to a seen patch, otherwise the model learns to use the short-cut.

Table 8: Changing masking strategy. Experiments trained for 300 ep. on IN-1k whilst using different masking strategies. Evaluated with frozen probe on IN-1k and ADE20K.

			IN-1k acc.			ADE20K mIoU		
Masking strategy	Seen (%)	Target (%)	Patch	CLS	X-Blk	kNN	Lin	Blk
Random (MAE-style)	25.0	75.0	2.6	2.5	10.9	2.3	1.4	4.3
Green noise (ColorMAE-style)	25.0	75.0	36.0	32.0	58.3	7.6	10.5	23.5
Inverse block (data2vec-style)	20.0	80.0	47.8	43.3	67.1	8.8	15.1	27.9
Cyclic block (CAPI/Franca)	25.0 (20–30)	75.0 (70–80)	67.0	65.8	73.7	20.8	25.1	32.2
Cyclic block (CAPI/Franca)	30.0 (25–35)	70.0 (65–75)	**68.0**	67.4	74.0	21.4	25.8	32.4
Multi-block (I-JEPA)	24.3 (9–37)	44.5 (24–64)	65.3	66.5	72.6	20.6	24.5	31.7
Multi-block (ours)	29.2 (13–43)	47.5 (26–71)	67.8	**68.9**	**74.4**	**22.1**	**26.6**	**34.2**

C.5 Masking hyperparameter sensitivity

We ablate the masking hyperparameters of Bootleg: the size of each mask region (Table 9), the number of mask regions (Table 10), and the aspect ratio range of mask regions (Table 11).

For each ablation, we report the average fraction of patch tokens which were seen by the encoder, and the fraction of patch tokens used as a target by at least one mask. We also show the fraction of targets which were adjacent to a seen token.

C.5.1 Mask size

Table 9 shows the effect of scaling mask region size while keeping K=4 regions fixed. The default mask scale (100%) yields the best results across all metrics. The performance falls off similarly as the mask size is increased or decreased. Smaller masks (80%) leave too much context visible (40.3% seen), reducing the difficulty of the prediction task, while larger masks (110–120%) occlude too much of the image, limiting the encoder's access to informative context. Performance holds out well across the range of mask sizes explored, despite the mean seen rate ranging from 22% to 40%.

Table 9: Effect of mask region size. The mask scale is varied as a percentage relative to the default (highlighted), keeping the number of mask regions fixed at $K=4$. ViT-S, 300 ep. on IN-1k.

					IN-1k acc.			ADE20K mIoU
K	Mask scale	Seen (%)	Target (%)	Adj. (%)	Patch	CLS	X-Blk	kNN
4	80% → [0.128, 0.146]	40.3 ± 7.7	40.6 ± 6.4	27.8 ± 10.2	65.3	64.9	73.2	19.7
4	90% → [0.144, 0.165]	33.9 ± 7.1	44.5 ± 7.0	21.7 ± 9.2	67.2	68.3	74.2	21.1
4	100% → [0.160, 0.183]	29.3 ± 6.7	47.3 ± 7.3	17.7 ± 8.3	**67.8**	**68.9**	**74.4**	**22.0**
4	110% → [0.176, 0.201]	25.5 ± 6.4	49.7 ± 7.6	14.5 ± 7.5	66.7	67.7	73.6	21.7
4	120% → [0.192, 0.220]	21.8 ± 6.4	52.2 ± 8.0	11.6 ± 6.8	65.1	66.2	73.1	21.2

C.5.2 NUMBER OF MASKS

Table 10 varies the number of mask regions, K, while adjusting their size to keep the total masked area approximately constant. The default $K=4$ achieves the best results, with $K=3$ and $K=5$ close behind. Fewer, larger masks ($K=1$, $K=2$) produce contiguous masked regions that occlude a large area, elevating the task. Many small masks ($K=8$) scatter targets across the image, reducing task difficulty slightly due to the reduced size of each mask. The results suggest that a moderate number of mask regions provides the best balance between prediction difficulty and spatial diversity. Performance holds out well across the range of number of mask regions.

Table 10: Effect of changing the number of mask regions, K. The mask scale is adjusted to keep the seen fraction of the image approximately constant. The mask scale percentage is relative to the default ($K=4$). ViT-S, 300 ep. on IN-1k.

					IN-1k acc.			ADE20K mIoU
K	Mask scale	Seen (%)	Target (%)	Adj. (%)	Patch	CLS	X-Blk	kNN
1	379% → [0.575, 0.726]	28.4 ± 8.0	64.8 ± 5.0	19.5 ± 7.1	67.2	67.5	73.5	21.2
2	208% → [0.281, 0.432]	29.1 ± 7.7	52.0 ± 8.0	16.1 ± 7.0	67.1	67.6	73.6	21.4
3	131% → [0.196, 0.254]	28.7 ± 7.2	47.3 ± 7.8	16.0 ± 8.0	67.5	68.6	74.0	21.7
4	100% → [0.160, 0.183]	29.3 ± 6.7	47.3 ± 7.3	17.7 ± 8.3	**67.8**	**68.9**	**74.4**	**22.0**
5	86% → [0.126, 0.168]	28.6 ± 7.4	48.9 ± 7.6	18.2 ± 9.0	67.3	68.1	74.1	21.8
6	72% → [0.109, 0.138]	30.0 ± 6.9	48.9 ± 7.1	20.8 ± 9.1	66.9	67.7	73.8	21.1
8	57% → [0.083, 0.111]	29.2 ± 7.0	50.8 ± 6.9	22.1 ± 9.6	65.5	66.1	73.5	20.4

C.5.3 MASK ASPECT RATIO

Table 11 compares different aspect ratio ranges for mask regions. The default range $[0.667, 1.5]$ achieves the best performance across all metrics. There is only a small impact on the performance of the model when changing the mask aspect ratio.

Using square masks slightly increases the difficulty in the task because more targets are further from seen components of the image. Similarly, wider aspect ratio ranges ($[0.5, 2]$ and $[0.333, 3]$) increases the task difficulty slightly as highly elongated mask rectangles allow the model to see context closer to the target patches. The default aspect ratio provides the best balance in task complexity, though the effect is not large.

Table 11: Effect of mask aspect ratio range. The default aspect ratio (highlighted) allows moderate variation; $[1, 1]$ constrains masks to be square. ViT-S, $K=4$, 300 ep. on IN-1k. Although the mask scale parameter is held constant, the seen rate varies due to changes in the distribution of possible mask areas for a given aspect ratio range.

				IN-1k acc.			ADE20K mIoU
Aspect ratio	Seen (%)	Target (%)	Adj. (%)	Patch	CLS	X-Blk	kNN
[1, 1]	25.1 ± 5.6	49.4 ± 7.5	14.0 ± 7.0	67.2	68.0	73.6	21.5
[0.667, 1.5]	29.3 ± 6.7	47.3 ± 7.3	17.7 ± 8.3	**67.8**	**68.9**	**74.4**	**22.0**
[0.5, 2]	30.1 ± 6.9	47.2 ± 7.2	18.8 ± 8.8	67.2	68.3	74.2	21.6
[0.333, 3]	30.4 ± 7.0	47.8 ± 7.2	19.8 ± 9.4	67.1	67.7	73.9	21.3