Exploring Target Representations for Masked Autoencoders

Xingbin Liu[1,2*] **Jinghao Zhou**[2*] **Tao Kong**[2*]
[1]Xiamen University [2]ByteDance

Abstract

Masked autoencoders have become popular training paradigms for self-supervised visual representation learning. These models randomly mask a portion of the input and reconstruct the masked portion according to assigned target representations. In this paper, we show that a careful choice of the target representation is unnecessary for learning good visual representation since different targets tend to derive similarly behaved models. Driven by this observation, we propose a multi-stage masked distillation pipeline and use a randomly initialized model as the teacher, enabling us to effectively train high-capacity models without any effort to carefully design the target representation. On various downstream tasks, the proposed method to perform masked knowledge **d**istillation with **bo**otstrapped **t**eachers (**dBOT**) outperforms previous self-supervised methods by nontrivial margins. We hope our findings, as well as the proposed method, could motivate people to rethink the roles of target representations in pre-training masked autoencoders. The code and pre-trained models are publicly available at https://github.com/liuxingbin/dbot.

1 Introduction

Masked **I**mage **M**odeling (MIM) [18, 34, 2, 40] has recently become an active research topic in the field of visual representation learning. To be specific, MIM randomly masks a portion of the input and then reconstructs the masked portion according to the transformed target, formulated as

$$\min_{\theta} \mathop{\mathbb{E}}_{x\sim\mathcal{D}} \mathcal{M}(\mathcal{T}(x \odot (1-M)), f_\theta(x \odot M)), \tag{1}$$

where "$\odot$" means element-wise product; M is the *patch mask*; "$x \odot M$" represents "unmasked patches" and vice versa; $f_\theta(\cdot)$ is the learnable network to be pre-trained; $\mathcal{T}$ is the transformation function generating the reconstructed target. $\mathcal{T}$ can either be a parameterized network or a traditional image feature transformation method; $\mathcal{M}(\cdot,\cdot)$ is the similarity measurement.

A crucial problem of MIM is how to choose the reconstructed target, *i.e.*, $\mathcal{T}(\cdot)$ in Eq. (1). Previous methods use disparate teacher networks to generate the reconstruction target [3, 34, 35, 18, 40]. Though different methods differ in their architectural designs and optimization, the choice of the teacher network lies crucial for each method and calls for a systematic study. In this work, we paraphrase a term **M**asked **K**nowledge **D**istillation (MKD) to focus our discussion on a special case of MIM where the target is generated by a parameterized network, *i.e.*, $\mathcal{T}(\cdot) = h_\phi(\cdot)$.

The purpose of our work is to investigate *whether a careful design of the teacher network for MKD matters*. To this end, we compare student networks distilled by four teacher networks with different computation pipelines [30, 18, 6, 31]. To our surprise, although the behaviors of the teacher networks are very different, the distilled student networks share similar characters after several stages of masked

[*] Equal contribution.
Work done during Xingbin's internship at ByteDance.

computation	initialized	classification				object detetion					semantic segmentation				
pipeline	teacher	0^{th}	1^{st}	2^{nd}	3^{rd}	0^{th}	1^{st}	2^{nd}	3^{rd}	4^{th}	0^{th}	1^{st}	2^{nd}	3^{rd}	4^{th}
Supervised	DeiT [31]	81.8	83.6	84.3	84.3	49.1	50.5	52.5	52.4	-	46.4	49.2	50.4	49.9	-
Contrastive	DINO [6]	83.2	84.2	84.5	84.4	50.1	52.5	52.9	52.7	-	46.8	49.7	50.4	49.4	-
Autoregressive	DALL-E [30]	81.1	83.5	84.4	84.3	31.9	51.0	52.7	52.5	-	31.9	47.4	49.6	49.3	-
Autoencoding	MAE [18]	83.6	84.3	84.4	84.3	50.6	52.9	52.7	52.5	-	48.1	49.6	50.4	49.8	-
-	random	77.3	83.4	84.5	84.3	29.2	49.6	52.4	52.7	52.4	25.7	47.0	49.1	49.5	49.5
performance variance		2.24	0.37	0.07	0.04	9.54	1.23	0.17	0.12	-	9.19	1.15	0.54	0.23	-

Table 1: The top-1 classification accuracy on ImageNet-1K, object detection AP-box on COCO with Cascade Mask R-CNN, and semantic segmentation mIoU on ADE20K with UperNet of dBOT using different models as the initialized teacher network. Note that all models are pre-trained on ImageNet-1K. We perform distillation in each stage for 800 epochs. In the 1^{st} stage, we distill from initialized teacher to obtain a student. In the subsequent (*i.e.*, 2^{nd}, 3^{rd}, etc.) stages, the obtained students are leveraged as bootstrapped teacher to distill a new student.

knowledge distillation. Such observations indicate that the design of target representation is not essential for learning good visual representations when pre-trained with multi-stage, *i.e.*, *teacher networks do not matter with multi-stage masked knowledge distillation.* Exceptionally, we use a randomly initialized model as teacher to perform multi-stage masked knowledge distillation, and find that it performs as well as those initialized by pre-trained models with the exact same settings! Using a *random model* as teachers not only avoids an extra pre-training stage, but also alleviates the painstaking selection of the target representations.

Based on the above studies and observations, we naturally propose to perform masked knowledge **d**istillation with **bo**otstrapped **t**eachers, short as **dBOT** 🤖. Specifically, masked knowledge distillation is performed repeatedly in multiple stages. At the end of each stage, we assign the student's weight to the teacher and re-initialize the student's weight to continue masked knowledge distillation. With simple yet effective design that enables pre-training starting from randomly initialized teachers, dBOT outperforms previous self-supervised methods by nontrivial margins on various downstream tasks.

2 Related work

Self-supervised learning is an active research topic recently. Early practices revolve around contrastive learning [19, 7, 17, 5, 6] where the model output features of images transformed by different data augmentations are pulled together. With the development of Masked Language Modeling (MLM) in language pre-training [12], researchers also introduce the training strategy of masked reconstruction to visual pre-training. BEiT [3] uses the DALL-E [30] to encode an image patch as the target for model reconstruction. iBOT [40] uses an online teacher shifting the target from offline to online to make the target semantic meaningful. In addition to using the token obtained from offline or online model as reconstruct target, MAE [18], and MaskFeat [34] achieve good performance in masked-image reconstruction using low-level pixels or HOG [10] features. However, there exists no work conferring a system-level study on the importance of how to choose adequate target representation.

3 Does $h_\phi(\cdot)$ Matter in MKD?

Given the general form of masked knowledge distillation as shown in Eq. (1), in this section, we aim to investigate *whether the careful design of the target,* i.e., *teacher network* $h_\phi(\cdot)$, *matters.* We employ the standard masked autoencoder framework [18] to give a system-level study.

Common setup. The architectural settings strictly follow [18]. For the teacher network, we use the vanilla ViT [13] with intact input. For the student network with masked input, we use the asymmetric encoder-decoder structure. The student's output is further projected to a dimension the same as that of teacher's embedding. During pre-training, we use Smooth L1 loss [14] for the optimization of the student network, and the teacher network is kept fixed. Detailed settings are delayed to Appendix C.1. We pre-train models on ImageNet-1K [11] and conduct evaluation under classification on ImageNet, object detection on COCO [24], and semantic segmentation on ADE20K [39].

3.1 Preliminary Study

We first investigate the effect of using networks initialized differently as teachers for masked knowledge distillation. Four canonical methods as ***pre-trained teachers*** are substantiated, *i.e.*, DeiT [31] for supervised learning, DINO [6] for contrastive learning, DALL-E [30] for autoregressive generation, and MAE [18] for autoencoding. The results of initialized teacher at the 0^{th} stage and of its distilled student at the 1^{st} stage are shown in Table 1.

Different $h_\phi(\cdot)$ lead to similarly performed students. After the first stage of masked knowledge distillation, the student consistently outperforms teacher as shown in Table 1. Although the performance order of different $h_\phi(\cdot)$ is reserved after the first stage of distillation, the students distilled from different $h_\phi(\cdot)$ have closer downstream performances compared to the original $h_\phi(\cdot)$. The performance variance drops from 2.24 to 0.37 after the first stage of distillation.

3.2 Distillation with Multiple Stages

Given the observations that better teacher generally induces better outperforming student, we are motivated to use the trained student as teacher to train new student repeatedly and study whether similar trend endures. If so, we would like to seek at what stage the performances saturate, as well as the discrepancy among the results incurred by different initialized teachers.

$h_\phi(\cdot)$ **does not matter with multi-stage distillation.** The performance gain is valid but decreases with multi-stage and eventually vanishes. Take MAE being the initialized teacher as an example, students outperform teachers by +0.7%, +0.1%, -0.1% for classification, from the 0^{th} to the 3^{rd} stage. Other teachers and downstream tasks share the same conclusion. Moreover, the performance gaps of students learned from different teachers decrease, especially after multi-stage, as shown by the performance variance at different stages in the last row of Table 1, which reveals that the choice of $h_\phi(\cdot)$ exerts little influence on the downstream performance. To demonstrate models' differences in terms of weights and outputs, we conduct a property analysis in Appendix B. Similar properties are found, which verify our conclusion.

A random $h_\phi(\cdot)$ works surprisingly well. Since the choice of $h_\phi(\cdot)$ does not matter, an intuitive experiment is to see what will happen when we employ a ***random teacher***, in which the parameters are randomly initialized at the 0^{th} stage. To our surprise, using a random teacher achieves performances comparably with other pre-trained teachers. The saturated results are on par with those induced by pre-trained teachers, which enables us to train a state-of-the-art model more efficiently, without the need of an extra pre-training stage for the initialized teacher (*e.g.*, contrastive learning as DINO).

4 MKD with Bootstrapped Teachers

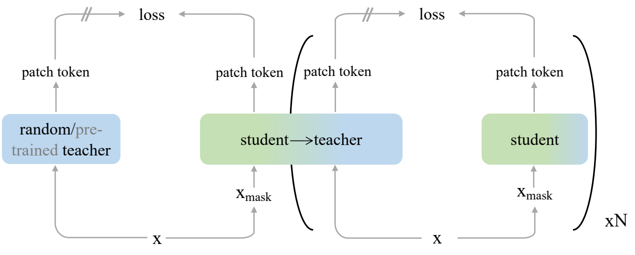

Figure 1: **Overview of dBOT**. dBOT uses a multi-stage distillation pipeline, *i.e.*, the parameters of the teacher network are frozen except at breakpoints, where we assign student parameters to the teacher and re-initialize the student network.

The study in Sec. 3 motivates us to propose a multi-stage distillation pipeline for pre-training. The entire pre-training undergoes multiple stages split by breakpoints. For each stage, we fix teacher network to obtain a stable visual representation, guiding the learning of student network. The pre-trained student model is then used as a stronger teacher and distills its knowledge to a new subsequent student, providing richer visual representations. We re-initialize the student network at each breakpoint. The above process repeats itself - the teachers keep bootstrapped from the students, until a performance saturation on downstream tasks is observed. Hence, our strategy is to perform distillation with *bootstrapped teacher*s. We illustrate our framework in Fig. 1.

5 Experiments

Architecture. We use different capacity Vision Transformers [13], *i.e.*, ViT-B/16, ViT-L/16, and ViT-H/14 for dBOT. The input image of size 224×224 is first divided by a linear projection head into non-overlapping patch tokens total of 196 for ViT-B and ViT-L, and 256 for ViT-H. We exactly follow the common setup demonstrated in Sec. 3, *e.g.*, a student with asymmetric encoder-decoder architecture, a teacher with intact input, etc.

Optimization. The learning rate is first linearly increased to the initial learning rate for the first 40 epochs and then cosine annealed to 0. The initial learning rate is set as 1.5e-4 × batch_size / 256, with batch size being 4096 for all models. We use the AdamW optimizer [26] and Smooth L1 loss [14] to optimize the parameters of student network. Stochastic drop rate are applied, 0.2 for ViT-B, 0.2 for ViT-L, and 0.3 for ViT-H. We use only center-crop and flipping for data augmentation. As shown in Table 1, the performance of different downstream tasks saturates at different stages. By default, we pre-train all models for classification with 2 stages, for object detection and semantic segmentation with 3 stages.

Evaluation setup. We sweep the base learning rate within a range with a batch size being 1024. We warm up the learning rate during the first 5 epochs to the initial learning rate and use a cosine schedule for the rest of the epochs. We average all the patch tokens output from the last transformer block and pass them into a linear projection head for classification. We fine-tune ViT-B for 100 epochs and ViT-L and ViT-H for 50 epochs in total.

Comparison with previous results. We report the fine-tuning results on ImageNet-1K, mainly focusing on the comparison of the self-supervised and supervised methods. Supervised denotes the results reported in the MAE. As shown in Table 2, dBOT achieves remarkable results with different model capacities, demonstrating its scalability. We achieved top-1 evaluation accuracy of 84.5%, 86.6%, and 87.4% with ViT-B, ViT-L, and ViT-H, yielding gains of 0.9%, 0.7%, and 0.5% compared to MAE. When fine-tuned with an image size of 448, dBOT further achieves an accuracy of 88.0%, surpassing the results obtained by MAE. More results can be found in Appendix A.

method	ViT-B	ViT-L	ViT-H	ViT-H$_{448}$
supervised [18]	82.3	82.6	83.1	-
MoCo v3 [8]	83.2	84.1	-	-
DINO [6]	83.6	-	-	-
methods based on masked image modeling:				
BEiT [3]	83.2	85.2	-	-
iBOT [40]	84.0	85.2	-	-
MAE [18]	83.6	85.9	86.9	87.8
data2vec [2]	84.2	86.2	-	-
dBOT	**84.5**	**86.6**	**87.4**	**88.0**

Table 2: Comparison result of the previous methods on ImageNet-1K. We evaluate by the end-to-end fine-tuning protocol. All results are based on an image size of 224, except for ViT-H with an extra result with 448 image size. We perform distillation in each stage for 800 epochs and with 2 stages (our default) in total.

6 Conclusion

As a special case of MIM, we formulate MKD upon which an empirical investigation is conducted about the influence of different target representations on self-supervised masked autoencoders. The study concludes that it is *not necessary* to carefully choose the target representation to learn good visual representations if distillation is performed in multiple stages (*i.e.*, with bootstrapped teachers). Instead of initializing teachers with pre-trained models, we resort to random ones for simple practice. *Without an extra stage of pre-training*, dBOT achieves favorable performance. We hope our study and method will provide timely insights for self-supervised learning.

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[2] 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 *ICML*, 2022.

[3] Hangbo Bao, Li Dong, and Furu Wei. BEiT: BERT pre-training of image transformers. In *ICLR*, 2022.

[4] Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: high quality object detection and instance segmentation. *TPAMI*, 2019.

[5] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *NeurIPS*, 2020.

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

[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.

[8] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *ICCV*, 2021.

[9] ED Cubuk, B Zoph, J Shlens, and Q Le Randaugment. Practical automated data augmentation with a reduced search space. In *CVPR Workshops*, 2020.

[10] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.

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

[14] Ross Girshick. Fast R-CNN. In *ICCV*, 2015.

[15] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *ICAIS*, 2010.

[16] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

[17] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. In *NeruIPS*, 2020.

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

[19] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

[20] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. In *NeurIPS Workshops*, 2015.

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

[22] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *ICCV Workshops*, 2013.

[23] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*, 2009.

[24] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[25] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *ICLR*, 2017.

[26] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

[27] Luke Melas-Kyriazi, Christian Rupprecht, Iro Laina, and Andrea Vedaldi. Deep spectral methods: A surprisingly strong baseline for unsupervised semantic segmentation and localization. In *CVPR*, 2022.

[28] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *ICVGIP*, 2008.

[29] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[30] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *ICML*, 2021.

[31] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers & distillation through attention. In *ICML*, 2021.

[32] 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 *CVPR*, 2018.

[33] Michael E Wall, Andreas Rechtsteiner, and Luis M Rocha. Singular value decomposition and principal component analysis. In *A practical approach to microarray data analysis*, 2003.

[34] Chen Wei, Haoqi Fan, Saining Xie, Chao-Yuan Wu, Alan Yuille, and Christoph Feichtenhofer. Masked feature prediction for self-supervised visual pre-training. In *CVPR*, 2022.

[35] Longhui Wei, Lingxi Xie, Wengang Zhou, Houqiang Li, and Qi Tian. MVP: Multimodality-guided visual pre-training. *arXiv preprint arXiv:2203.05175*, 2022.

[36] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *ECCV*, 2018.

[37] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, 2019.

[38] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018.

[39] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, 2017.

[40] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. iBOT: Image bert pre-training with online tokenizer. In *ICLR*, 2021.

method	AP^{box}		AP^{mask}	
	ViT-B	ViT-L	ViT-B	ViT-L
supervised [40]	49.8	51.2	43.2	44.5
DINO [6]	50.1	-	43.4	-
MAE [18]	50.6	54.0	43.9	46.2
iBOT [40]	51.3	-	44.3	-
dBOT	**52.7**	**56.0**	**45.7**	**48.2**

Table A1: Object detection and instance segmentation results on COCO using Cascade Mask R-CNN. All results are based on our implementation with the official pre-trained model. We perform distillation in each stage for 800 epochs and with 3 stages (default).

method	mIoU		mAcc	
	ViT-B	ViT-L	ViT-B	ViT-L
supervised [18]	47.4	49.9	-	-
iBOT [40]	48.4	52.3	59.3	63.3
data2vec[2]	48.2	-	59.5	-
MAE [18]	48.1	53.6	58.9	65.5
dBOT	**49.5**	**54.5**	**60.7**	**66.0**

Table A2: Semantic segmentation results on ADE20K using UperNet. All results are based on our implementation with the official pre-trained model. We perform distillation in each stage for 800 epochs and with 3 stages (default).

method	Cif_{10}	Cif_{100}	iNa_{18}	iNa_{19}	Flwrs	Cars	avg.
sup. [40]	99.0	90.8	73.2	77.7	98.4	92.1	88.5
DINO [6]	99.1	91.7	72.6	78.6	98.8	93.0	89.0
iBOT [40]	99.2	92.2	74.6	79.6	98.9	94.3	89.8
MAE [18]	-	-	75.4	80.5	-	-	-
dBOT	**99.3**	91.3	**77.9**	**81.0**	98.2	93.7	**90.2**

Table A3: Transfer classification accuracy on various datasets. We report the results of ViT-B. Sup. denotes the supervised baseline. The average results (avg.) are shown in the rightmost column.

A More Experiments

A.1 Downstream Tasks

To further demonstrate the effectiveness, we consider dense prediction tasks: object detection, semantic segmentation, and instance segmentation, as well as classification tasks that transfer to smaller datasets.

Objection detection and instance segmentation. We consider Cascade Mask R-CNN [4] as the task head for object detection and instance segmentation with ViT-B and ViT-L on COCO [24]. We report AP^{box} and AP^{mask} for object detection and instance segmentation respectively. The results are demonstrated in Table A1. dBOT outperforms the previous self-supervised and supervised methods by a large margin, setting a new state-of-the-art result with both ViT-B and ViT-L. With ViT-B, dBOT achieves a AP^{box} of 52.7 and a AP^{mask} of 45.7, outperforming the supervised baseline pre-training by 2.9 and 2.5 points, respectively. With ViT-L, such improvement is more prominent with 4.8 and 3.6 points respectively, showing the high scalability of dBOT for model capacity in downstream dense prediction tasks.

Semantic segmentation. We adapt UperNet [36] as the task head for semantic segmentation with ViT-B and ViT-L on ADE20K [39]. We report the mIoU and mAcc for semantic segmentation, and the results are demonstrated in Table A2. We achieve the best performances on semantic segmentation compared to previous self-supervised methods by a nontrivial margin. dBOT improves mIoU from 47.4 to 49.5 with ViT-B, and 49.9 to 54.5 with ViT-L, yielding gains of 2.1 and 4.6 points respectively, compared to the supervised baseline. The improvement in semantic segmentation is as significant as in object detection.

Transfer learning. To further investigate the generalizability of visual representations learned by dBOT. We study transfer learning performance by fine-tuning the pre-trained models on smaller datasets, including CIFAR10 [23] (Cif_{10}), CIFAR100 [23] (Cif_{100}), iNaturalist18 [32] (iNa_{18}), iNaturalist19 [32] (iNa_{19}), Flowers [28] (Flwrs), and Cars [22]. The results are shown in Table A3. dBOT achieves comparable, if not better, performances compared to previous best methods. Specifically, the improvement is significant on relatively larger datasets like iNaturalist18 and iNaturalist19, with 4.7% and 3.3% respectively compared to the supervised baseline.

pre-training epochs	acc
1600	83.6
800-800	**84.5**
533-533-533	84.4

(a) **Stage split number**. 2-stage distillation works the best.

pre-training epochs	acc
400-800	84.3
800-400	84.3
800-800	**84.5**
800-1200	84.3

(b) **Epoch for each stage**. 2-stage distillation with 800 epochs for each stage works the best.

momentum	acc
vanilla	**84.5**
0.9998	83.6
0.9999	83.9
cosine(0.996,1)	82.1

(c) **Momentum update**. The *vanilla* strategy explicitly splitting stages works the best.

target norm	acc
w/ [LN]	84.3
w/o [LN]	**84.5**

(d) **Target normalization**. Using patch representations w/o [LN] as targets works the best.

student init	acc
w/o re-initialize	84.2
w/ re-initialize	**84.5**

(e) **Student initialization**. Re-initializing the student's weight at breakpoints works best.

mask ratio	acc
0.7	84.3
0.75	**84.5**
0.8	84.2

(f) **Mask ratio**. A mask ratio of 75% works best.

Table A4: **Ablation study** with **ViT-B/16** on ImageNet-1K validation set. We report with the end-to-end fine-tuning top-1 accuracy (%). Ablation study is conducted with randomly initialized teachers. We note that models distilled from the pre-trained teachers generally share similar trends. Default settings are marked in gray . *vanilla* denotes m being 0 at the breakpoint and 1 otherwise. cosine(a,b) denotes m is cosine annealed from value a to b.

A.2 Ablation Study

Stage split number. We study the influence of stage number by splitting total training epochs of 1600 into varying distillation stages, from 0 to 2. Results are shown in Table A4a. 2-stage distillation works the best (for classification task), achieving 84.5% accuracy. Splitting epochs to 3-stage brings 0.1% performance drop, while all splitting strategies obtain a top-1 accuracy higher than 83.6%, indicating its generalizability.

Epoch for each stage. Table A4b studies proper epochs needed for each stage in a 2-stage distillation pipeline. With the 2nd stage distilling for 800 epochs, longer epochs for the 1st stage induces 0.2% improvement (84.3% *vs.* 84.5%). With the 1st stage distilling for 800 epochs, 800 epochs are enough for the 2nd stage since 1200 epochs incur no gain. Evenly splitting the epochs in 2-stage masked knowledge distillation achieves the best performance.

Momentum update. We use in dBOT a multi-stage distillation pipeline, which is to distill from a momentum encoder with m being 0 for every breakpoint and 1 otherwise. We further investigate other momentum update strategies commonly used in self-supervised learning. Results are shown in Table A4c. The *vanilla* strategy works the best.

Target normalization. We study whether patch tokens obtained by the self-attention blocks to be used as target representation should be passed through the Layer Normalization [1] layer [LN]. The accuracy of models after 2-stage distillation is shown in Table A4d. Without passing through [LN], the patch tokens directly obtained from the transformer block make them less suitable as target representations to guide students' learning.

Student initialization. We study whether student's weight should remain when entering the next stage of distillation. Specifically, we either keep the student's weight unchanged or re-initialize the student at each breakpoint. As shown in Table A4e, re-initializing the student's weight works the best.

Mask ratio. Table A4f shows the influence of the mask ratio on end-to-end fine-tuning. The optimal mask ratio for dBOT is 75%, the same as that in MAE.

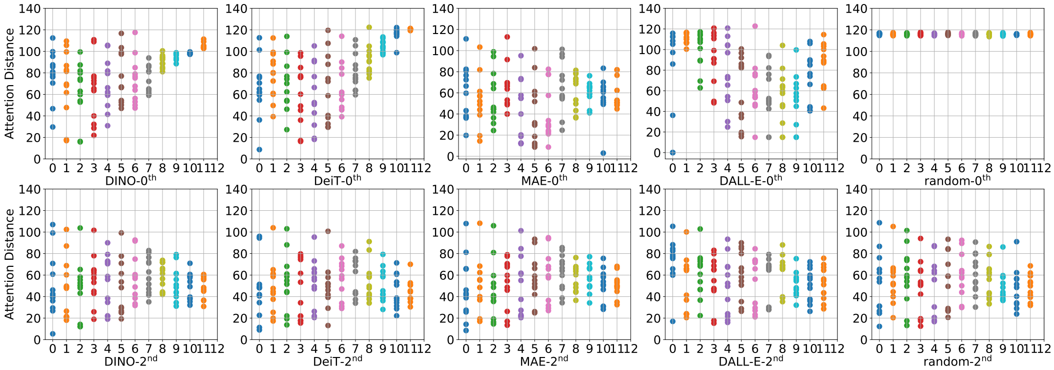

Figure B1: Average attention distance of different heads w.r.t layer number of ViT-B with different distilling teachers and their corresponding student distilled for 2 stages. The first row showcases the teachers while the second showcases the 2^{th} stage distilled student. Models using different teachers achieve the same result. The distilled students obtain comparatively more local attention compared to the teachers.

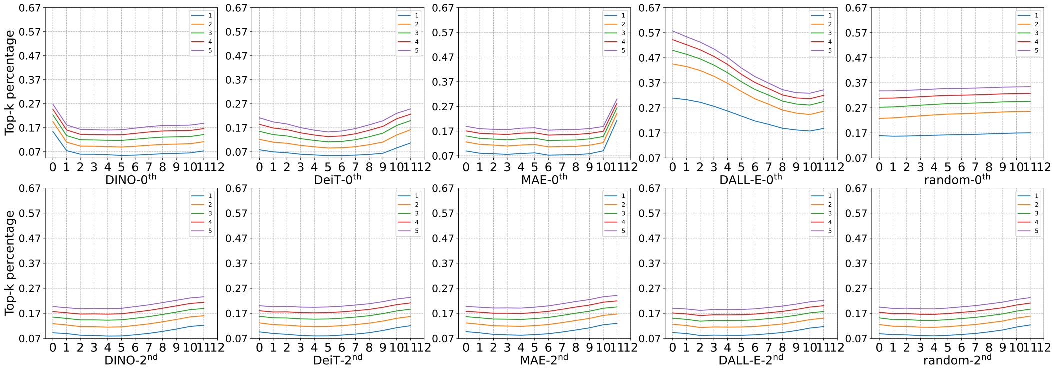

Figure B2: Singular value decomposition of different layers of ViT-B with different distilling teachers and their corresponding student distilled for 2 stages. The first row showcases the teachers while the second showcases the 2^{th} stage distilled student. Models using different teachers achieve the same result.

B Property Analysis

We investigate the properties of models distilled from different teachers under certain criteria, analyzing models' weights and outputs. Further, training efficiency is briefly discussed with previous methods.

Averaged attention distance. We compute averaged attention distance [13], averaged over ImageNet-1K val set, for each attention head of different blocks to understand how local and global information flows into Transformers. Average attention distance for dBOT using DeiT, DINO, MAE, DALL-E, and random as teachers are illustrated in Fig. B1. The higher the attention distance, models' attention over an image is more global. Although the average attention distance of disparate initialized teachers varies greatly, their distilled students after multi-stage distillation exhibit similar behaviors, *e.g.*, models' attention toward local or global contents. Additionally, dBOT achieves more local attention than previous works.

Singular value decomposition. We computed the percentage of top-k singular values [33] of the embedding w.r.t each layer. The results are averaged over the ImageNet-1K val set. We showcase the results with k varying from 1 to 5. Singular value decomposition for dBOT using DeiT, DINO, MAE, DALL-E, and random as teachers are shown in Fig. B2. The higher the percentage, the models' output over an image is less correlated, indicating larger redundancy of its spatial representations

model	DeiT	DINO	DALL-E	MAE	random
0th	13.8	36.1	36.5	36.6	23.6
2nd	36.6	36.6	36.6	36.6	36.6

Table B5: The results of unsupervised object detection on Pascal VOC 2012 with CorLoc based on SVD decomposition.

method	data2vec [2]	BEiT [3]	MAE [18]	dBOT
asym.	✗	✗	✓	✓
ViT-B	169	166	79	109
ViT-L	431	356	125	200
ViT-H	960	751	240	416

Table B6: Training time (s) per epoch for different methods with ViT-B/16, ViT-L/16, and ViT-H/14. *asym.* denotes whether to use an asymmetric encoder-decoder structure [18]. All entries are tested on the same setting, *i.e.*, with 32 NVIDIA A100-80G GPUs.

thus less suitability for compression. Intuitively, random models at the 0th stage has the largest percentage given that pixel are merely randomly projected. The student networks distilled from different initialized teachers exhibit similar behaviors.

Unsupervised object detection. We use unsupervised object localization to quantitatively evaluate the visual representation obtained by different models. We follow the evaluation practice proposed in [27] with Correct Localization (CorLoc) on POC-VOC 2012 trainval sets, except that we conduct feature decomposition via SVD instead of Laplacian since we observe more stable behaviors with SVD. We first compute singular value decomposition for the patch feature obtained by the ViT-B last block. Then a sign operation is applied on the first eigenvector, obtaining a binary mask of an image. We then take the bounding box around the largest connected component, which is more like the foreground object instead of the background. Correct localization (CorLoc) is used to measure the results, evaluated on POC-VOC 2012 trainval sets. A box is considered to have correctly identified an object if it has more than 50% intersection-over-union with a ground truth bounding box. Quantitative results are demonstrated in Table B5. dBOT using different teachers achieves very similar results, with students consistently outperforming their teachers.

Training efficiency. We compute the training time per epoch for different methods in Table B6. With an asymmetric encoder-decoder architecture (*asym.*) as the default setup, dBOT performs slower than MAE, but much faster than data2vec and BEiT. Such advantage turns more significant with models of larger size.

C Implementation Details

C.1 Pre-Training

Default setup. We show our *default* pre-training setup in the second colum of Table C7. We use Xavier Uniform [15] to initialize the Vision Transformer [13]. Note that we use asymmetry stochastic drop path rate for students and teachers.

Setup for distillation from bigger teachers. We follow the *default* setup, except that we use a different setup for stages. We first train larger-size teachers for 2 stages (in all downstream tasks) and use those to distill new students for 1 stage (in all downstream tasks).

C.2 Classification

The *default* end-to-end fine-tuning recipe is shown in the second column of Table C8, following the common recipes [18, 3] of ViT tuning for self-supervised models. The same recipe is applied when distilling from bigger teachers.

config	default	recipe⚗
optimizer	AdamW [26]	
optim. momentum β_1	0.9	
optim. momentum β_2	0.95	0.98
loss	Smooth L1	negative cos.
peak learning rate	2.4e-3	3e-3
learning rate schedule	cosine decay [25]	
batch size	4096	
weight decay	0.05	
stages	2 (c.), 3 (d./s.)	1
epochs per stage	800	1600
warmup epochs [16]	40	10
augmentation	RandomResizedCrop	
aug. input scale	(0.2, 1)	(0.4, 1)
asym. enc-dec [18]	✓	✗
drop path [21]	0.2 (B/L), 0.3 (H)	0.1 (B/L/H)
target w/ `[LN]`	✗	✓
mask ratio	0.75	0.4

Table C7: **Pre-training setup.** recipe⚗ is the pre-training recipe for dBOT⚗ . cos. denotes cosine distance. c., d., and s. denotes downstream tasks of classification, object detection, and semantic segmentation respectively. drop path is for the students.

config	default	recipe⚗
optimizer	AdamW [26]	
peak learning rate	{0.8,1.2,1.6,2}e-3	{1,2,3,4}e-4
weight decay	0.05	
optim. momentum	$\beta_1, \beta_2 = 0.9, 0.999$	
layer-wise decay	0.75	
batch size	1024	
learning schedule	cosine decay	
warmup epochs	5	
epochs	100 (B), 50 (L/H)	
augmentation	RandAug (9, 0.5) [9]	
label smoothing	0.1	
mixup [38]	0.8	
cutmix [37]	1.0	
drop path [21]	0.2 (B/L), 0.3 (H)	0.1 (B), 0.2 (L), 0.3 (H)

Table C8: **End-to-end fine-tuning setup.** recipe⚗ is the pre-training recipe for dBOT⚗ .

C.3 Object Detection and Instance Segmentation

We adopt the vanilla ViT with Cascade Mask R-CNN [4] as the task head on COCO [24] dataset for object detection and instance segmentation, following the common setup [40]. The default recipe is shown in Table C9. To cope with versatile image sizes, we add relative position embedding instead of interpolating the absolute position embedding obtained during pre-training. For a fair comparison, we applied the same setup and sweep the learning rate and stochastic drop path rate for different methods.

C.4 Semantic Segmentation

We use vanilla ViT and UperNet [36] as the task head on ADE20K [39] dataset for semantic segmentation, following the common setup [3]. The default recipe is shown in Table C10. To cope with versatile image sizes, we add relative position embedding instead of interpolating the absolute position embedding obtained during pre-training. For a fair comparison, we applied the same setup and sweep the learning rate and layer-wise decay for different methods.

config	value
optimizer	AdamW [26]
optim. momentum	$\beta_1, \beta_2 = 0.9, 0.999$
peak learning rate	1e-4
batch size	16
layer-wise decay	0.75
weight decay	0.05
learning schedule	step
epochs	12
step epochs	8, 11
drop path [21]	0.2

Table C9: **Object detection setup.**

config	value
optimizer	AdamW [26]
optim. momentum	$\beta_1, \beta_2 = 0.9, 0.999$
peak learning rate	{0.3,0.5,0.8,1,3}e-4
batch size	16
layer-wise decay	{0.65,0.75,0.85.0.95}
weight decay	0.05
learning schedule	cosine
steps	16000
warmup steps	1500
drop path [21]	0.1(B), 0.2(L)

Table C10: **Semantic segmentation setup.**

D Additional Experiments

D.1 Pixels *vs.* Random Mapping of Pixels

epoch	pixel	0^{th} block	12^{th} block
400	83.3	83.2	83.2
1600	83.6	83.6	83.6

MAE performs masked image modeling using the image pixel as the reconstruction target. We directly alter the target to patch tokens obtained from the image fed into a randomly initialized network. We select two patch tokens as the reconstruction target, one is the token obtained using the last transformer block, and the other is the token obtained using linear projection, *i.e.*, without any transformer block. After 400 epoch pre-training of ViT-B, the top-1 accuracy of the model on ImageNet-1K obtained by the three different targets is shown below.

It can be derived that using the patch token obtained by a randomly initialized network as the target can achieve comparable results with a pixel as a target. A similar result proves that patch tokens obtained by a randomly initialized can also serve as a good reconstruction target.

D.2 Object Detection with Mask R-CNN

Additionally, we use Mask R-CNN structure with FPN for object detection and instance segmentation on COCO datasets. The results are shown in Table D11. dBOT outperforms other methods by a large margin, which is similar to the results using Cascade Mask R-CNN.

D.3 Linear Probing

We evaluate the linear probing performance of dBOT and MAE using ViT-B following the same setup as MAE, the results of which is 67.8% and 67.9% respectively. dBOT achieves comparable linear probing performances with MAE.

E Distill from Bigger Teachers

Inspired by canonical practices in knowledge distillation [20], we use larger teachers to distill smaller students, showcasing the potential of MKD in general. Specifically, we attempt to use ViT-L/H as teacher networks to distill ViT-B, and ViT-H as the teacher network to distill ViT-L. All larger teachers are first distilled for 2 stages with the default setup. We resize the image to 196×196 for ViT-H/14 to keep the length of its output the same as that of ViT-B/L. While we do not find substantial gains on classification results, the results by distilling from ViT-H are significantly better for dense prediction tasks compared to the default setup, *i.e.*, +0.8 points of AP^{box} and +1.3 points of mIoU with ViT-B as the student. The performance gain in distilling ViT-L from ViT-H is diminished but still valid, *i.e.*, +0.1 AP^{box} and +0.7 mIoU. We also consider MKD with data-richer teachers, *e.g.*CLIP, as exploratory experiments and set new state-of-the-art results for self-supervised learning. Refer to Appendix F for details.

method	AP^{box}		AP^{mask}	
	ViT-B	ViT-L	ViT-B	ViT-L
supervised	47.9	49.3	42.9	43.9
iBOT [40]	48.6	50.6	43.1	44.7
data2vec [2]	41.1	46.1	37.0	41.0
MAE [18]	50.2	53.5	44.8	47.4
dBOT	**51.4**	**54.0**	**45.8**	**48.0**

Table D11: Object detection and instance segmentation results on COCO using **Mask R-CNN**. We report the result both with ViT-B and ViT-H. All results are based on our implementation with official released pre-trained model.

teacher	student	cls.	det.	seg.
ViT-B	ViT-B	84.5	52.7	49.5
ViT-L		84.6 (+0.1)	53.1 (+0.4)	50.1 (+0.6)
ViT-H		84.6 (+0.1)	53.5 (+0.8)	50.8 (+1.3)
ViT-L	ViT-L	86.6	56.0	54.5
ViT-H		86.8 (+0.2)	56.1 (+0.1)	55.2 (+0.7)

Table E12: Results of classification (cls.) on IN1K, object detection (det.) on COCO, and semantic segmentation (seg.) on ADE20K. For same-size teachers (colored gray), students are pre-trained with default settings. For bigger teachers, students are pre-trained for 1-stage from 2-stage distilled teachers.

F Distill from Data-Richer Teachers

We explore to use models pre-trained with richer data (*i.e.*, CLIP [29] with 400M Image-Text pairs) as the initialized teacher to seek a potential upper-bound of MKD.

F.1 Pre-Training

Compared to the *default* setup, there exist two major disparities of the pre-training recipes for models distilled from data-richer teachers, discussed next. The following practice is summarized as *recipe*⚗ detailed in Table C7.

Vanilla Architecture. We find that not using the asymmetric encoder-decoder architecture [18] is optimal, as shown in Table F13. While an asymmetric architecture generates momentum for bootstrapping models similar to [17], which lies crucial for distillation with random teachers, it hurts the performance when distilling with stronger pre-trained teachers.

Hypothetically, the significance of the decoder in asymmetrical encoder-decoder architecture lies in the need for separate layers to decode low-level details when the targets contain little semantics (*e.g.*, pixels and random mappings of pixels). Such a need is eased when the target contains high-level semantics (*e.g.*, DINO and CLIP). The existence of the decoder, in this case, may even restrain the encoder to grasp full knowledge from the teacher, inducing degraded performances.

1-Stage MKD. We use different models as teachers to distill students for one stage with longer epochs, *i.e.*, 1600. Results are shown in Table F14. Empirically, the performance gains for multi-stage MKD over 1-stage MKD decrease as teachers' fine-tuning performance increases. Stronger teachers, such as DINO and MAE, induce similarly performed students with 1-stage MKD (1×1600) compared to 2-stage MKD (2×800).

Specifically, when using CLIP as the pre-trained teacher, the performance for 2-stage MKD is, to our surprise, 0.9% lower than that of 1-stage MKD. Understandably, although the fine-tuning result of the student after 1-stage distillation is better than that of CLIP, the student is essentially trained on IN1K and may not contain faithfully data information stored in the CLIP model. Therefore, strong teachers work well with 1-stage MKD, especially for models pre-trained on extra richer data.

initialized teacher	pre-training data	asym. enc-dec	acc
random	IN1K	✓	84.5
random	IN1K	✗	83.8
DINO [6]	IN1K	✓	84.4
DINO [6]	IN1K	✗	84.8
CLIP [29]	IN1K + 400M ITp.	✓	84.9
CLIP [29]	IN1K + 400M ITp.	✗	85.7

Table F13: Image classification on IN1K with DINO and CLIP as initialized teachers, as well as random ones. Students with DINO and CLIP as teachers are distilled for 1 stage.

pre-training epochs	random	DALL-E[30]	DeiT[31]	DINO[6]	MAE[18]	CLIP[29]
0	77.3	81.1	81.8	83.2	83.6	84.8
1×1600	83.6	83.6	83.6	84.4	84.4	84.9
2×800	84.5	84.4	84.3	84.5	84.4	84.0
△	+0.9	+0.8	+0.7	+0.1	+0.0	-0.9

Table F14: ImageNet-1K classification results of 1 stage masked knowledge distillation with different teachers. Total epochs are shown in the format of (stages×epochs_per_stage). △ denotes performance gaps between entries of 2×800 and 1×1600.

F.2 Downstream Tasks

Implementation Details. For fine-tuning, we also use a slightly different recipe from *default* one with smaller learning rates and drop path, dubbed as *recipe*⚗ detailed in Table C8. For object detection, instance segmentation, and semantic segmentation, we follow the default setup detailed in Appendices C.3 and C.4.

Results. Results for downstream tasks are shown in Table F15. ViT-B distilled from CLIP-B achieves an 85.7% top-1 accuracy and a 52.9 mIoU, surpassing all previous arts. With CLIP-L as the teacher, ViT-H with image resolution 448 achieves an **89.1%** top-1 accuracy, setting a new state-of-the-art image recognition result.

F.3 Conflict with Main Conclusion

It can be observed that MKD with CLIP [29] as the teacher performs much better than that with the random teacher and multi-stage distillation, which seems contradictory to our main conclusion that *teacher networks do not matter with multi-stage masked knowledge distillation.* Notably, CLIP is trained with 400M image text pairs (300× larger than ImageNet-1K), which is a drastically different setup from multi-stage distillation on ImageNet-1K only. Exploring CLIP as a target representation gains popularity [35] recently but is beyond the main scope of this paper. We present these results to corroborate the validity and to explore the upper bound of MKD in general. We note that the exact solution to resolve the conflict is to perform multi-stage distillation using the CLIP's in-house 400M data to which we have no access. It is hypothesized that two results should be matched in light of experiments on ImageNet-1K, which is left to future work.

initialized teacher	student	cls.	det.	seg.
random	ViT-B	84.5	52.7	49.5
CLIP-B [29]		85.7 (+1.2)	53.6 (+0.9)	52.9 (+3.4)
random	ViT-L	86.6	56.0	54.5
CLIP-L [29]		87.8 (+1.2)	56.8 (+0.8)	56.2 (+1.7)
random	ViT-H	87.4	-	-
CLIP-L [29]		88.5 (+1.1)	-	-

Table F15: Results of classification (cls.) on IN1K, object detection (det.) on COCO, and semantic segmentation (seg.) on ADE20K with CLIP [29] as the teacher. Students are distilled for 1 stage. The det. results with CLIP as teachers are with absolute positional embedding.