Fuse Before Transfer: Knowledge Fusion for Heterogeneous Distillation (Appendix)

Guopeng Li[1*] Qiang Wang[3] Ke Yan[3] Shouhong Ding[3] Yuan Gao[2†] Gui-Song Xia[2†]
[1]School of Computer Science, [2]School of Artificial Intelligence, Wuhan University
[3]Tencent YouTu Lab
{guopengli, guisong.xia}@whu.edu.cn, ethan.y.gao@gmail.com
{albertqwang, kerwinyan, ericshding}@tencent.com

A. Implementation Details

For training models of various architectures on the ImageNet-1K and CIFAR100 datasets, we use different optimization settings and hyperparameters for CNN and MSA/MLP students following the code and paper of OFA [6]. The detailed settings can be found in Tab. 1. Our code and models are from Timm library [20]. Given that the training pipeline for VisionMamba [22] is currently not integrated into the timm library [20], we don't consider it following OFA [6].

Besides, we set γ in $\mathcal{L}_{\mathrm{OFA}}$ as 1.0 on CIFAR100 [9] and 1.5 in ImageNet-1K [4] like OFA [6]. The τ_2 is learnable in $\mathcal{L}_{\mathrm{InfoNCE}}$. The weight of $\mathcal{L}_{\mathrm{OFA}}$ and $\mathcal{L}_{\mathrm{InfoNCE}}$ are equal. We averagely divide the model into 4 stages if the model is not originally 4-stage in this paper.
If the student and teacher are heterogeneous models, the logits of our fused model is:

$$p_{\mathrm{f}}(x) = fc_{\mathrm{m}} \circ \mathrm{S}_{\mathrm{m}}^{4} \circ \overbrace{(\mathrm{MSA} \circ \mathrm{PE})}^{\mathrm{L2G}} \circ \mathrm{S}_{\mathrm{c}}^{3} \circ \mathrm{S}_{\mathrm{c}}^{2} \circ \mathrm{S}_{\mathrm{c}}^{1}(x), \quad (1)$$

where x is the input image, $\mathrm{S}_{\mathrm{c}}^{i}$ denotes CNN modules, L2G includes patch embedding module (PE) and multi-head-self-attention block (MSA), $\mathrm{S}_{\mathrm{m}}^{i}$ denotes MSA/MLP modules, and fc_{m} denotes the fully-connected layers of MSA/MLP models. Note that our fused model is also CNN-MSA/MLP connection when the teacher is CNN model. When the CNN teacher is frozen, the first three MSA/MLP stages learn to align with the first three CNN stages, and the last MSA/MLP stage learns to align with the teacher's output.

If the student and teacher are homogeneous (*i.e.*, both CNN/MSA/MLP models), the fused model is:

$$p_{\mathrm{f}}(x) = fc_{\mathrm{t}} \circ \mathrm{S}_{\mathrm{t}}^{4} \circ \overbrace{(\mathrm{MSA} \circ \mathrm{PE})}^{\mathrm{L2G}} \circ \mathrm{S}_{\mathrm{s}}^{3} \circ \mathrm{S}_{\mathrm{s}}^{2} \circ \mathrm{S}_{\mathrm{s}}^{1}(x), \quad (2)$$

where x is the input image, $\mathrm{S}_{\mathrm{s}}^{i}$ denotes student models, L2G includes patch embedding module (PE) and multi-head-self-attention block (MSA), $\mathrm{S}_{\mathrm{t}}^{i}$ denotes teacher models, and fc_{t} denotes the fully-connected layers of teacher models. We argue that homogeneous model pairs also have gaps in inductive bias and module functions, so the fusion works for them too in Tab. 3 of our main paper.

B. Comparisons with other methods

We compare the differences between some similar methods and our FBT in Fig. 1. Firstly, to the best of our knowledge, our FBT is the first work to bridge heterogeneous model pairs with knowledge fusion, which provides more flexible designs for heterogeneous knowledge fusion and transfer. Secondly, our fused model bridges the representation gaps between cross-architecture students and teachers by combining different inductive biases and module functions, making our FBT more suitable for cross-architecture distillation. Thirdly, as demonstrated in [6], the $\mathcal{L}_{\mathrm{OFA}}$ enhances the target information and hinders the transfer of incorrect information from the teacher by a modulating parameter γ [6], which is more suitable than $\mathcal{L}_{\mathrm{KL}}$ in cross-architecture distillation. Lastly, the $\mathcal{L}_{\mathrm{MSE}}$ aligns the features in a pixel-by-pixel manner, which is not reasonable for spatially different heterogeneous features, *e.g.* (A) and (E) in Fig. 2. Thus, as demonstrated in Tab. 4, we get the better performance by smoothing the features in spatial and apply contrastive learning by $\mathcal{L}_{\mathrm{InfoNCE}}$ to align the feature embeddings of cross-architecture models.

B.1 Comparisons with FCFD [11]

There are some works to input the student features to teachers in similar-architecture distillations, *e.g.*, ReviewKD [3], FCFD [11], and so on [2]. However, they are all designed for teacher-student pairs with similar architectures, suggesting different motivations and designs compared to our FBT for cross-architecture distillation. For example, FCFD [11]

	CIFAR100		ImageNet-1K	
	CNN	MSA/MLP	CNN	MSA/MLP
Epochs	100	300	300	300
Image resolution	224^2	224^2	224^2	224^2
Batch size	512	1024	1024	512
Initial LR	0.1	5e-4	0.1	5e-4
Minimum LR	1e-6	1e-6	1e-3	1e-5
Optimizer	SGD	AdamW	SGD	AdamW
Weight decay	1e-4	5e-2	2e-3	5e-2
LR schedule	$\times$0.1 at [30,60,90]	Cosine	Cosine	Cosine
Warmup	3	20	3	20
EMA	-	0.99996	-	-
RandAugment	-	9/0.5	-	9/0.5
Mixup	-	0.8	-	0.8
Cutmix	-	1.0	-	1.0
RE prob	-	0.25	-	0.25

Table 1. **Details of optimization settings.** The settings are following OFA [6].

has the best performance and is the most similar method to our FBT, but some important designs of our FBT are very different from FCFD.

Firstly, FCFD [11] is designed for CNN students and teachers, which needs to be modified seriously if we apply it to heterogeneous distillations. Besides, as demonstrated in Tab. 2, FCFD is not suitable for any cross-architecture teacher-student models. But our FBT is generic for any teacher-student pair.

Table 2. **Results of FCFD in cross-architecture distillations on CIFAR100 dataset.** As shown, FCFD is not suitable for cross-architecture distillations compared to our FBT.

Methods	T.	S.	T.	S.	T.	S.
	Swin-T	ResNet18	ViT-S	ResNet18	ConvNeXt-T	Swin-P
FCFD	78.34		53.58		77.29	
Our FBT	**81.61**		**81.93**		**80.34**	

Secondly, although FCFD [11] also combines different module functions, the connections between students and teachers are **random and mutual**, which makes it hard to converge to the optimal spaces and brings huge training costs, especially for cross-architecture distillations. Conversely, our FBT considers that the CNN models are feature extractors and the MSA/MLP models are feature aggregators [14], so the fused model is the CNN-MSA/MLP model and obeys the rule of "alternately replacing Conv blocks with MSA blocks from the end of a baseline CNN model" in [14]. For example, FCFD [11] includes the multiply random connections of MSA/MLP-CNN models (the first parts are MSA/MLP modules, and the latter parts are CNN modules) when we modify it to cross-architecture distillation. However, MSA/MLP-CNN models are unreasonable for the hybrid models [14], leading to bad distillation performance.

Thirdly, FCFD [11] is a two-level paradigm that only considers the knowledge transfer between the teacher and student, not introducing the knowledge transfer between the fused model and student. However, the supervision of the fused model is very important as demonstrated in our ablation study of the main paper.

Fourthly, FCFD [11] is designed for CNN models and does not consider the representation gaps between different inductive biases between cross-architecture models. Thus, the feature projectors of FCFD are CNN modules, which perform worse than our L2G modules because L2G includes the MSA modules to convert the local features to global receptive fields.

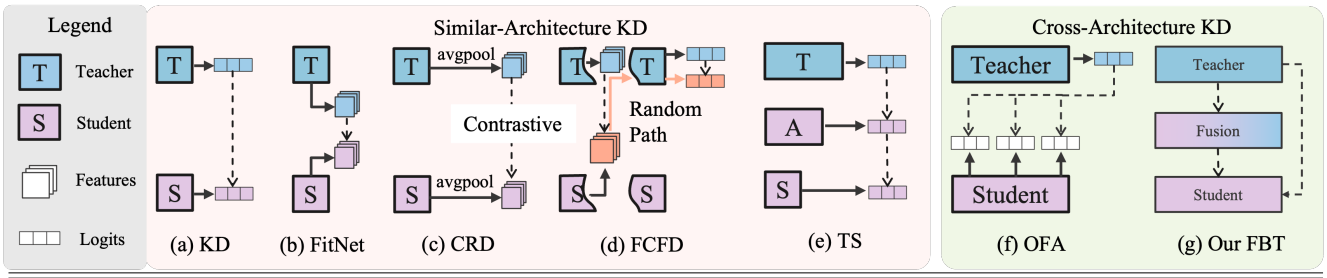
Lastly, FCFD [11] does not consider the gaps between different representation spaces of cross-architecture models. Thus, the loss functions of FCFD [11], *i.e.*, $\mathcal{L}_{\mathrm{KL}}$ and $\mathcal{L}_{\mathrm{MSE}}$ are not appropriate for cross-architecture teacher-student pairs. For example, as demonstrated in Tab. 4, $\mathcal{L}_{\mathrm{MSE}}$ is not suitable for some cross-architecture teacher-student pairs.

Experimentally, as shown in Tab. 3 of our main paper, our FBT has a competitive performance compared with FCFD [11] in similar-architecture distillations. More importantly, our FBT is generic for cross-architecture distillations, but FCFD is hard to achieve it in Tab. 2.

B.2 Comparisons with TS [13]

Although TS [13] introduces an assistant to bridge the gaps between students and teachers, it is designed for CNN models, which is unsuitable for our heterogeneous distillation.

Firstly, TS [13] is a multi-step distillation, which transfers the knowledge from the teacher to the assistant, and then from the assistant to the student. However, our FBT is a one-step distillation, which jointly transfers the knowledge among the teacher, fused model, and student. As shown in



	KD [8]	FitNet [15]	CRD [16]	FCFD [11]	TS [13]	OFA [6]	Ours
Knowledge to Distill	Logits	Feature	Feature	Feature	Logits	Logits	**Feature**
Generic	Yes	No	Yes	No	No	Yes	**Yes**
Scheme	T.-S.	T.-S.	T.-S.	T.-S.	T. $\rightarrow$ A. $\rightarrow$ S.	T.-S.	**T.-F.-S.**
Heterogeneous Inductive Bias Fusing	No	No	No	No	No	No	**Yes**
Heterogeneous Module Merging	No	No	No	Yes	No	No	**Yes**
Training Cost	**Very Low**	Middle	High	High	Middle	Middle	Low
Loss Function	$\mathcal{L}_{\mathrm{KL}}$	$\mathcal{L}_{\mathrm{KL}} + \mathcal{L}_{\mathrm{MSE}}$	$\mathcal{L}_{\mathrm{InfoNCE}}$	$\mathcal{L}_{\mathrm{KL}} + \mathcal{L}_{\mathrm{MSE}}$	$\mathcal{L}_{\mathrm{KL}}$	$\mathcal{L}_{\mathrm{OFA}}$	$\mathcal{L}_{\mathrm{OFA}} + \mathcal{L}_{\mathrm{InfoNCE}}$

Figure 1. **The taxonomy of our method.** Our methods are feature-based, generic, and three-level, which fuses heterogeneous inductive biases and module functions with an efficient fused model. Target-wise $\mathcal{L}_{\mathrm{OFA}}$ and spatial-agnostic $\mathcal{L}_{\mathrm{InfoNCE}}$ are more suitable for CAKD than $\mathcal{L}_{\mathrm{KL}}$ and $\mathcal{L}_{\mathrm{MSE}}$. To the best of our knowledge, our FBT is one of the pioneer works in feature-based generic distillation.

Tab 4 (E), the knowledge transfer between teacher and student is necessary, but TS [13] ignores that.

Secondly, progressive distillation [13] is a distilling strategy, not an algorithm discussed in Appendix D, which is orthogonal with our FBT.

Lastly, the only difference between the teacher, assistant, and student in TS [13] is the depth of layers. In other words, it is simple and designed for only CNN distillation, which is hard to apply in heterogeneous distillation. However, our FBT is adaptive for knowledge fusion between heterogeneous model pairs.

B.3 Comparisons with recent work [13]

More recently, [21] introduce contrastive distillation in heterogeneous distillation. But they also use some common nature (*e.g.*,low pass filter) to distill cross-architecture features. Differently, our FBT has the adaptive fused model for different model pairs, fusing heterogeneous inductive bias and module functions.

C. heterogeneous features

Fig. 2 shows heterogeneous features, demonstrating some important observations in our main paper.

Firstly, heterogeneous models have different inductive biases. For example, CNN models [7] have the inductive bias of "locality", thereby making the features local like (A-B) in Fig. 2. Differently, the features of MSA and MLP models [5, 12, 17] are global because of their global inductive bias, *e.g.*, (C-F) in Fig. 2. Therefore, combining different inductive biases mitigates the gaps between heterogeneous models like our fused model.

Secondly, heterogeneous models have different module functions. For example, the architectures/functions of ResNet [7] and Swin models [12] are hierarchical. They gradually expand the receptive fields and upsample the features, *e.g.*, (A-D) in Fig. 2. Differently, MLP models [17] and most MSA models [5] are uniform. The features of shadow and deep layers have higher similarity than the hierarchical CNN, *e.g.*, (E-F) in Fig. 2. Therefore, combining different module functions mitigates the gaps between heterogeneous models like our fused model.

Thirdly, features of heterogeneous models have different spatial distributions in different channels. For example, the different channels of CNN models have similar spatial localizations (such as the right figures of Fig. 2 (A-B)). Conversely, the features of MSA and MLP models in different channels are more diverse, *e.g.*, the right figures of Fig. 2(C-F). Besides, as demonstrated in [7, 14, 17], spatial smoothing is useful for the predictions of CNN/MSA/MLP models (*e.g.*, average pooling). Therefore, we smooth the features and replace pixel-by-pixel $\mathcal{L}_{\mathrm{MSE}}$ with $\mathcal{L}_{\mathrm{InfoNCE}}$ in our main paper. Tab. 4 also demonstrates the strength of applying $\mathcal{L}_{\mathrm{InfoNCE}}$ to smoothing features.

Methods (all methods only use the $\mathcal{L}_{\mathrm{KD}}$)	CIFAR100
(A) Swin $\rightarrow$ ResNet34 + Resnet34 $\rightarrow$ ResNet18	78.53
(B) Swin-fusion-ResNet18 (ours w/o $\mathcal{L}_{\mathrm{FBT}}(\mathrm{K_t}, \mathrm{K_s})$)	79.26
(C) Swin-fusion-ResNet18(ours w/ $\mathcal{L}_{\mathrm{FBT}}(\mathrm{K_t}, \mathrm{K_s})$	79.28
(D) Swin $\rightarrow$ ResNet18 + ResNet34 $\rightarrow$ ResNet18	80.07
(E) Swin-fusion-ResNet18 + ResNet34-fusion-ResNet18	81.24

Table 3. **Different distillation paradigm.** Swin denotes the Swin-Tiny model. Our method is the one-stage joint-optimization teacher-fusion-student paradigm, which is orthogonal with progressive distillation like [1, 13].

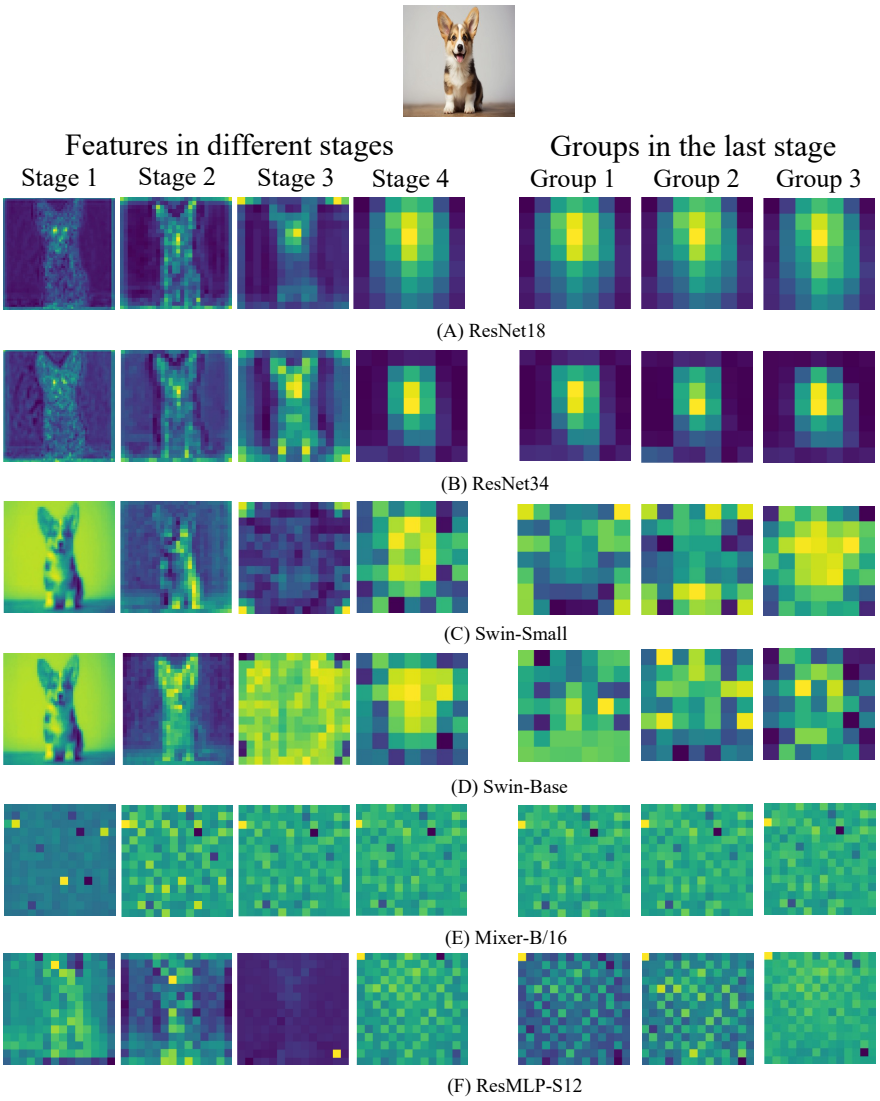


Figure 2. **Diverse features in different models.** The left figures are features in different stages (all models are divided into 4 stages). The right figures are the final features in different groups (we divide the channels of final features into 3 groups). The spatial distribution of features is diverse according to the channels, stages, and model architectures/functions.

D. Different distilling strategy

Multi-teacher progressive distillation is a training strategy [13] and our FBT is a training algorithm. They are **orthogonal** and can be used together. As shown in (D-E) in Tab. 3, we can replace the T.-S. with our T.-F.-S. paradigm to improve the results in each stage of progressive distillation. Besides, using multi-teacher distillation, we can improve the performance of a given student by applying our FBT to both SAKD and CAKD.

Loss	Top-1 accuracy
$\mathcal{L}_{\text{MSE}}$ in all intermediate features	24.06
$\mathcal{L}_{\text{MSE}}$ in the final features	65.17
$\mathcal{L}_{\text{MSE}}$ in the final features after average pooling	76.79
$\mathcal{L}_{\text{InfoNCE}}$ in the final features after average pooling	**78.01**

Table 4. **FitNet [15] with $\mathcal{L}_{\text{MSE}}$ *vs.* $\mathcal{L}_{\text{InfoNCE}}$ loss on CIFAR100.** The teacher is ConvNeXt-T (88.41% Top-1 accuracy) and the student is Swin-P (72.63% Top-1 accuracy) on CIFAR100. As shown, the smoothing features and $\mathcal{L}_{\text{InfoNCE}}$ are more suitable for cross-architecture distillations than the original features and $\mathcal{L}_{\text{MSE}}$.

MSA Block	T.	S.	T.	S.	T.	S.
	Swin-T	ResNet18	ViT-S	ResNet18	ConvNeXt-T	Swin-P
ViT Block	81.05		80.74		**80.72**	
Swin Block	**81.61**		**81.93**		80.34	

Table 5. **Our results with different MSA blocks.** T. and S. denote the teacher and the student. ViT block is from ViT [5] and Swin block is from Swin [12]. As shown, different blocks have different functions in different teacher-student pairs.

Teacher	Student	Student Params	Student FLOPs	OFA Branch		Our Branch	
				Params	FLOPs	Params	FLOPs
DeiT-T	ResNet18	11.69 M	1.82 G	4.92 M	0.1 G	**0.39 M**	**0.07 G**
ResNet50	DeiT-T	5.68 M	1.08 G	5.81 M	0.25 G	**0.54 M**	**0.10 G**
ConvNeXt-T	ResMLP-S12	15.32 M	3.01 G	21.64 M	0.99 G	**1.48 M**	**0.29 G**

Table 6. **Training cost.** The extra parameters of our FBT are about one-tenth of OFA [6]. The best results are **bold**.

E. Loss function for heterogeneous distillation.

In Tab. 4, we compare the results of FitNet with different settings on the CIFAR100 dataset. Firstly, the accuracy improves from 24.06 to 65.17 when we apply $\mathcal{L}_{\text{MSE}}$ only to features of the final stage, rather than intermediate stages. This demonstrates the intermediate features are not suitable for feature alignment in some cross-architecture teacher-student pairs. Secondly, the accuracy improves from 65.17 to 76.79 when we apply average pooling to features of the final stage. This demonstrates the diverse spatial distributions of features are not suitable for feature alignment in some cross-architecture teacher-student pairs. Thirdly, the accuracy improves from 76.79 to 78.01 when we replace $\mathcal{L}_{\text{MSE}}$ with $\mathcal{L}_{\text{InfoNCE}}$. This is because $\mathcal{L}_{\text{InfoNCE}}$ considers the releationships between different channels, but $\mathcal{L}_{\text{MSE}}$ is pixel-by-pixel.

F. L2G in our fused model.

As shown in Tab. 5, when we replace the Swin block [12] with ViT block [5] in our L2G, the performance on different teacher-student pairs has different rises and falls. Thus, the MSA block in L2G is also important and is worth exploring in future works.

Our L2G includes a patch embedding for dimension alignments of features and an MSA block for global information exchange.

Why do we use a patch embedding? The feature shape of a CNN model with the size (N, C, H, W), while that of an MSA/MLP model is denoted as (N, L, D). N indicates the batch size, and C, H, and W refer to the channel, height, and width of the CNN model's feature map respectively. L and D denote the patch number and embedding dimension of the ViT/MLP model's feature map. In our fused model, to connect the features of CNN and MSA/MLP models, we need to transform the feature map of the CNN model into the MSA/MLP-style (shape) feature through a "patchify" operation. Besides, the effectiveness of "divide image to patches" has been demonstrated in CNN models [19], MSA models [5, 12], and MLP models [10, 17, 18]. Therefore, we use a patch embedding to process the CNN features. More importantly, patch embedding is a kind of spatial smoothing that is beneficial to align heterogeneous features.

Why do we use an MSA block? The extra MSA block converts the local CNN features to a global receptive field, which is more suitable to input the later MSA/MLP models. Besides, the later MSA/MLP models are frozen when the teacher is the MSA/MLP model and the student is the CNN model. In this case, a learnable MSA block plays an important role in aligning heterogeneous features.

G. Training cost.

Beyond performance considerations, training cost is critical for the distillation. We compare the training cost of the recent OFA [6] and our FBT framework in Tab. 6. In Fig 2 of our main paper, OFA uses four extra feature projectors, but our FBT only uses one L2G to project the features at the third stage of CNN models. Therefore, as shown in Tab. 6, we introduce much fewer additional parameters and FLOPs on par with OFA [6] under different combinations of teacher and student models. Specifically, the number of extra parameters is about one-tenth that of OFA when the student and teacher are different architectures. As a result, our FBT is more efficient than OFA [6].

References

[1] Shengcao Cao, Mengtian Li, James Hays, Deva Ramanan, Yu-Xiong Wang, and Liangyan Gui. Learning lightweight object detectors via multi-teacher progressive distillation. In *International Conference on Machine Learning*, pages 3577–3598. PMLR, 2023. 3

[2] Defang Chen, Jian-Ping Mei, Hailin Zhang, Can Wang, Yan Feng, and Chun Chen. Knowledge distillation with the reused teacher classifier. In *Proceedings of the IEEE/CVF*

Conference on Computer Vision and Pattern Recognition (CVPR), 2022. 1

[3] Pengguang Chen, Shu Liu, Hengshuang Zhao, and Jiaya Jia. Distilling knowledge via knowledge review. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 1

[5] 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 *International Conference on Learning Representations (ICLR)*, 2021. 3, 5

[6] Zhiwei Hao, Jianyuan Guo, Kai Han, Yehui Tang, Han Hu, Yunhe Wang, and Chang Xu. One-for-all: Bridge the gap between heterogeneous architectures in knowledge distillation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 1, 2, 3, 5

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3

[8] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 3

[9] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 1

[10] Jiachen Li, Ali Hassani, Steven Walton, and Humphrey Shi. Convmlp: Hierarchical convolutional mlps for vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 5

[11] Dongyang Liu, Meina Kan, Shiguang Shan, and Xilin Chen. Function-consistent feature distillation. *International Conference on Learning Representations (ICLR)*, 2023. 1, 2, 3

[12] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 3, 5

[13] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant. In *Proceedings of the AAAI conference on artificial intelligence (AAAI)*, 2020. 2, 3, 4

[14] Namuk Park and Songkuk Kim. How do vision transformers work? In *International Conference on Learning Representations (ICLR)*, 2021. 2, 3

[15] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In *International Conference on Learning Representations (ICLR)*, 2015. 3, 4

[16] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. In *International Conference on Learning Representations (ICLR)*, 2020. 3

[17] Ilya O. Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, and Alexey Dosovitskiy. Mlp-mixer: An all-mlp architecture for vision. *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 3, 5

[18] Hugo Touvron, Piotr Bojanowski, Mathilde Caron, Matthieu Cord, Alaaeldin El-Nouby, Edouard Grave, Gautier Izacard, Armand Joulin, Gabriel Synnaeve, Jakob Verbeek, et al. Resmlp: Feedforward networks for image classification with data-efficient training. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2022. 5

[19] Asher Trockman and J Zico Kolter. Patches are all you need? *arXiv preprint arXiv:2201.09792*, 2022. 5

[20] Ross Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019. 1

[21] Hongjun Wu, Li Xiao, Xingkuo Zhang, and Yining Miao. Aligning in a compact space: Contrastive knowledge distillation between heterogeneous architectures. *arXiv preprint arXiv:2405.18524*, 2024. 3

[22] Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model, 2024. 1