Supplementary Materials

Seungbum Hong, Jihun Yoon, and Min-Kook Choi
VisionAI, hutom
Seoul, Republic of Korea
{qbration21, jhyoon2020, mkchoi}@hutom.io

Junmo Kim
KAIST
Daejeon, Republic of Korea
junmo.kim@kaist.ac.kr

Abstract

This material includes additional experimental results and architecture details of the Self-supervised knowledge transfer (SSKT). For PyTorch implementation, please refer to the following link[1].

1. SSKT with Multiple Tasks and Different Problem Domains

Self-supervised knowledge transfer (SSKT) supports a training procedure that enables effective transfer learning in a variety of scenarios using deep neural networks. SSKT has a structure that transfers pretrained knoweldge naturally, without compromising the training information of the pretrained network or requiring additional supervision in the target task training process. We achieved this goal using the soft label-based knowledge transfer techniques with auxiliary learning through self-supervision, for the various domain of image recognition variants. Final formulation of the SSKT as follows:

$$\begin{aligned}\underset{\theta_t}{\text{argmin}} \Big(& L(h_t^{prim}(x_i;\theta_t.D_t,T_t), y_{t,i}^{prim}) \\ & +\alpha(L(h_t^{aux}(f_{s_1}(x_i);\theta_t.D_t,T_{s_1}), y_{s_1,i}^{aux}) \\ & +L(h_t^{aux}(f_{s_2}(x_i);\theta_t.D_t,T_{s_2}), y_{s_2,i}^{aux}) + \ldots \\ & +L(h_t^{aux}(f_{s_M}(x_i);\theta_t.D_t,T_{s_M}), y_{s_M,i}^{aux}))\Big). \end{aligned} \quad (1)$$

We define a multi-task network $h_t(x;\theta_t,D_t,T_t)$, where x is the input, θ_t is a parameter of the target network, D_t is a target dataset, and T_t is the task to be trained. θ_t is updated simultaneously through target loss and auxiliary loss during training to solve the primary task. $h_s(x;\theta_s,D_s,T_s)$ denotes a source network that receives the input x and delivers knowledge to the target network. θ_s denotes a parameter trained by the source task T_s for the source data set D_s. θ_s is not updated during the target task training. i is the i^{th} batch of the training data, α is balanced parameter for total loss, and $y_{s,i}^{aux} = h_s(x_i;\theta_s,D_s,T_s)$ is the softmax output from the pretrained source network and conveys the dark knowledge of the pretrained dataset by soft labels. The data transformation function f_s converts the data type to match the source task to infer the recognition information to the task of the source domain. For example, if T_t is an action recognition problem using 3D-CNN, the input $x^{w\times h\times d} \in D_t$ is defined as a three-dimensional tensor. In this case, if a pretrained network for transfer learning is obtained through the image recognition problem T_s using 2D-CNN, $f_s : x^{w\times h\times d} \rightarrow \hat{x}^{w\times h}$ should be defined as a function that maps a three-dimensional tensor to a two-dimensional matrix into which h_s can be input. Up to M number of different type of transformation functions could be defined. Algorithm1 describes how the SSKT works depending on each transfer learning scenario.

Transfer Modules Depending on CNN Architecture. To encourage predicting $y_{s,i}^{aux}$ by h_t, we design bottleneck structure based transfer module supporting auxiliary task using feature output from each convolutional block. Figure 1 shows configuration of transfer modules depending on each CNN architecture for its problem domain. We applied the transfer module to four different CNN architectures such as ResNet [5], DenseNet [6], MobileNet [10], and 3D-ResNet [4] for each problem domain.

2. Additaional Experiments Results

We provide performance for all experimental configurations for each dataset, in addition to the results contained in the main manuscript. For fair comparison of SSKT, the configurations consist of a combination of the type of source and target network, the presence or absence of a transfer module, and a loss function. For model architecture and hyperparameters setting for training (See Table 1 of the main manuscript). Same as the experiment results of the main manuscript, the datasets of the source task are ImageNet [2] and Places365 [13], and the datasets of the target task are CIFAR10/100 [7], STL10 [1], ImageNet, Places365, PAS-

[1] https://github.com/generation21/generation6011

Algorithm 1: The SSKT algorithm

```
***Input*** : Target Network($\theta_t$), Source Network($\theta_s$),
 Dataset($D_{t,s}$), Task ($T_{t,s}$), Transfer Module ($TM$)
***Initialization*** : Learning rate=$\alpha$, $j = 0$
**while** *convergence* **do**
    **else if** *SSKT with Single Source* **then**
        Update
        $\theta_{j+1} \leftarrow$
        $\theta_j - \alpha\nabla_{\theta_j}(\mathcal{L}(h_t^{prim}(x_{(i)};\theta_t.D_t,T_t),y_{(t,i)}^{prim}) +$
        $\mathcal{L}(TM(h_s^{aux}(x_{(i)};\theta_t.D_t,T_s),y_{(s,i)}^{aux})))$
    **else if** *SSKT with Multiple Sources* **then**
        Update
        $\theta_{j+1} \leftarrow$
        $\theta_j - \alpha\nabla_{\theta_j}(\mathcal{L}(h_t^{prim}(x_{(i)};\theta_t.D_t,T_t),y_{(t,i)}^{prim}) +$
        $\sum_{m=1}^{M}\mathcal{L}(TM(h_{s_m}^{aux}(x_{(i)};\theta_t.D_t,T_{s_m}),y_{(s_m,i)}^{aux})))$
    **else if** *SSKT with Single Source with Different
     Problem Domains* **then**
        Update
        $\theta_{j+1} \leftarrow$
        $\theta_j - \alpha\nabla_{\theta_j}(\mathcal{L}(h_t^{prim}(x_{(i)};\theta_t.D_t,T_t),y_{(t,i)}^{prim}) +$
        $\mathcal{L}(TM(h_s^{aux}(f_s(x_{(i)});\theta_t.D_t,T_s),y_{(s,i)}^{aux})))$
    **else if** *SSKT with Multiple Sources with
     Different Problem Domains* **then**
        Update
        $\theta_{j+1} \leftarrow$
        $\theta_j - \alpha\nabla_{\theta_j}(\mathcal{L}(h_t^{prim}(x_{(i)};\theta_t.D_t,T_t),y_{(t,i)}^{prim}) +$
        $\sum_{m=1}^{M}\mathcal{L}(TM(h_{s_m}^{aux}(f_s(x_{(i)});\theta_t.D_t,T_{s_m}),y_{(s_m,i)}^{aux})))$
**end**
```

CAL VOC [3], UCF101 [12], and HMDB51 [8]. Tables 1 to 6 provide performance according to the experimental conditions of each dataset. Figure 2 shows the performance changes for the STL10 and PASCAL VOC datasets depending on hyperparameters, the structure of the source and target network, and the presence or absence of a transfer module. The abbreviations for the datasets and model architectures listed in all experimental tables are as follows:

Datasets: ImageNet (I), Places365 (P), CIFAR10 (C10), CIFAR100 (C100), STL10 (S10), PASCAL VOC (VOC), UCF101 (U101), and HMDB51 (H51).

Model architectures: ResNet (R), DenseNet (D), MobileNetV2 (MV2), and 3D-ResNet (3DR).

Finally, we included the experimental results according to the training setting for further analysis of the SSKT. Table 7 shows the all the combination of comparison results for DenseNet121 and MoblieNetV2, and Table 8 shows the evaluations for each experimental setting with finetuning scenario. Table 9 shows the performance comparison with MAXL [9] which is the state-of-the-art self-supervised learning based on auxiliary learning.

Table 1. Performance change according to the configuration of the SSKT for CIFAR10 dataset compared to the training from scratch. All experiments evaluated test performance 3 times from the same random seed for the model. TM stands for Transfer Module and R[depth] stands for ResNet structure. The best performance of each network architecture highlighted in **bold**.

T_s	T_t	Model	Method	TM	Loss	acc.
-	C10	R20	scratch	-	CE	92.19±0.09
P		R20	SSKT	x	CE+CE	92.21±0.06
		R20	SSKT	x	CE+KD	92.24±0.14
		R20	SSKT	o	CE+CE	92.23±0.04
		R20	SSKT	o	CE+KD	92.25±0.04
I		R20	SSKT	x	CE+CE	92.28±0.07
		R20	SSKT	x	CE+KD	92.34±0.07
		R20	SSKT	o	CE+CE	92.44±0.05
		R20	SSKT	o	CE+KD	92.29±0.0
P+I		R20	SSKT	x	CE+CE	91.9±0.1
		R20	SSKT	x	CE+KD	**92.46±0.15**
		R20	SSKT	o	CE+CE	92.42±0.07
		R20	SSKT	o	CE+KD	92.22±0.17
-		R32	scratch	-	CE	93.21±0.09
P		R32	SSKT	x	CE+CE	92.77±0.14
		R32	SSKT	x	CE+KD	92.87±0.31
		R32	SSKT	o	CE+CE	92.65±0.26
		R32	SSKT	o	CE+KD	92.59±0.22
I		R32	SSKT	x	CE+CE	93.26±0.08
		R32	SSKT	x	CE+KD	92.78±0.2
		R32	SSKT	o	CE+CE	93.25±0.12
		R32	SSKT	o	CE+KD	92.88±0.07
P+I		R32	SSKT	x	CE+CE	92.88±0.15
		R32	SSKT	x	CE+KD	93.07±0.09
		R32	SSKT	o	CE+CE	**93.38±0.02**
		R32	SSKT	o	CE+KD	93.1±0.22

References

[1] A. Coates, A. Ng, and H. Lee. An analysis of single-layer networks in unsupervised feature learning. In *In Proc. of AISTAT*, 2011. 1

[2] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *In Proc. of CVPR*, 2009. 1

[3] M. Everingham, S. M. A. Eslami, L. V. Gool, C. K. I. Williams, Winn J, and A. Zisserman. The pascal visual object classes challenge - a retrospective. *International Journal of Computer Vision*, 111(1):98–136, 2015. 2

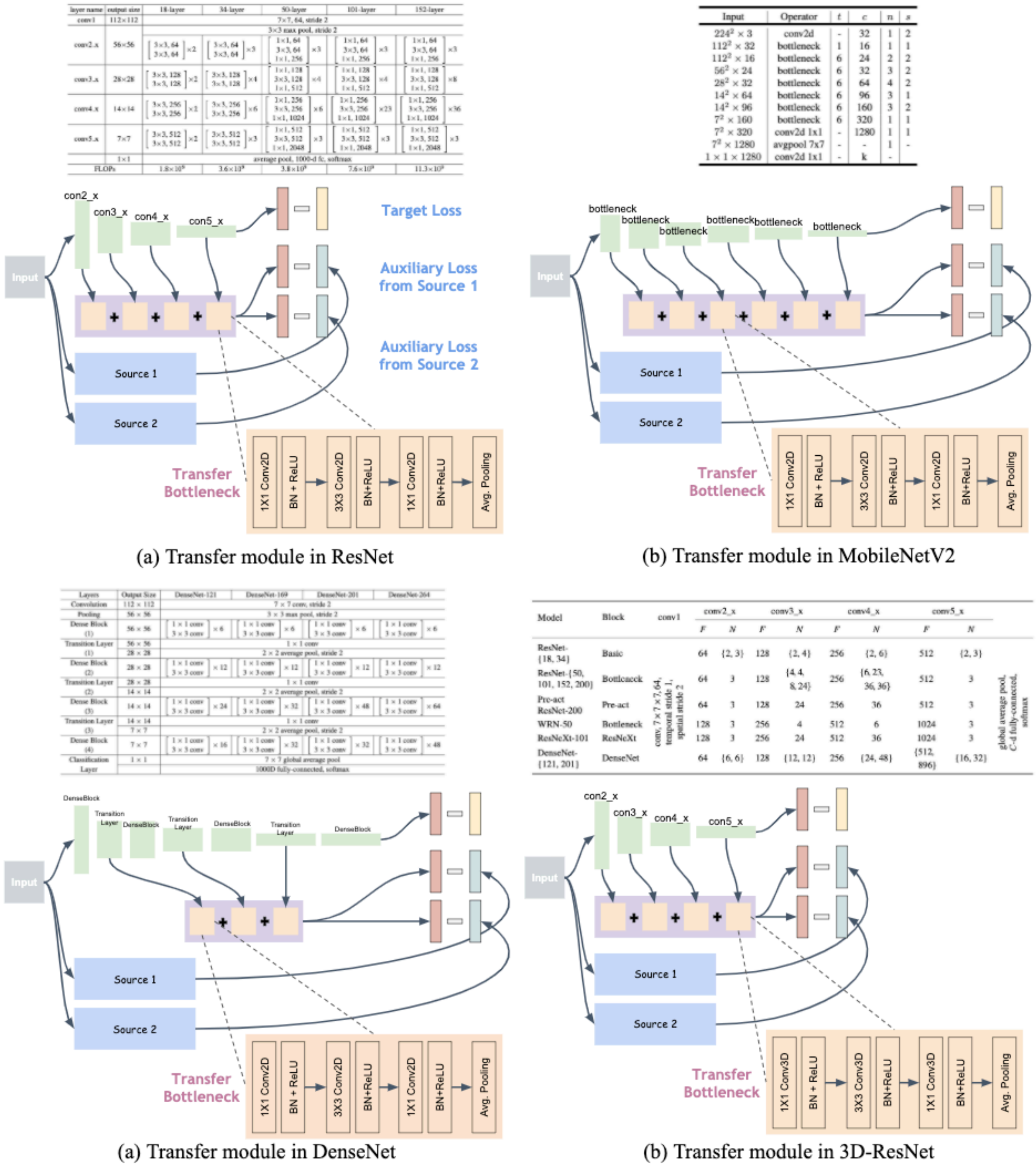
[4] K. Hara, H. Kataoka, and Y. Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *In Proc. of CVPR*, 2018. 1

[5] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *In Proc. of CVPR*, 2016. 1

[6] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *In Proc. of CVPR*, 2017. 1

[7] A. Krizhevsky. Learning multiple layers of features from tiny images. In *Technical Report*, 2009. 1

[8] H. Kuehne, H. Jhuang, E. Garrote, T. A. Poggio, and T. Serre. Hmdb: a large video database for human motion recognition. In *In Proc. of ICCV*, 2011. 2



layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		[3×3, 64 3×3, 64] ×2	[3×3, 64 3×3, 64] ×3	[1×1, 64 3×3, 64 1×1, 256] ×3	[1×1, 64 3×3, 64 1×1, 256] ×3	[1×1, 64 3×3, 64 1×1, 256] ×3
conv3_x	28×28	[3×3, 128 3×3, 128] ×2	[3×3, 128 3×3, 128] ×4	[1×1, 128 3×3, 128 1×1, 512] ×4	[1×1, 128 3×3, 128 1×1, 512] ×4	[1×1, 128 3×3, 128 1×1, 512] ×8
conv4_x	14×14	[3×3, 256 3×3, 256] ×2	[3×3, 256 3×3, 256] ×6	[1×1, 256 3×3, 256 1×1, 1024] ×6	[1×1, 256 3×3, 256 1×1, 1024] ×23	[1×1, 256 3×3, 256 1×1, 1024] ×36
conv5_x	7×7	[3×3, 512 3×3, 512] ×2	[3×3, 512 3×3, 512] ×3	[1×1, 512 3×3, 512 1×1, 2048] ×3	[1×1, 512 3×3, 512 1×1, 2048] ×3	[1×1, 512 3×3, 512 1×1, 2048] ×3
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

(a) Transfer module in ResNet

Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	

(b) Transfer module in MobileNetV2

Layers	Output Size	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	112 × 112	7 × 7 conv, stride 2			
Pooling	56 × 56	3 × 3 max pool, stride 2			
Dense Block (1)	56 × 56	[1 × 1 conv 3 × 3 conv] × 6	[1 × 1 conv 3 × 3 conv] × 6	[1 × 1 conv 3 × 3 conv] × 6	[1 × 1 conv 3 × 3 conv] × 6
Transition Layer (1)	56 × 56	1 × 1 conv			
	28 × 28	2 × 2 average pool, stride 2			
Dense Block (2)	28 × 28	[1 × 1 conv 3 × 3 conv] × 12	[1 × 1 conv 3 × 3 conv] × 12	[1 × 1 conv 3 × 3 conv] × 12	[1 × 1 conv 3 × 3 conv] × 12
Transition Layer (2)	28 × 28	1 × 1 conv			
	14 × 14	2 × 2 average pool, stride 2			
Dense Block (3)	14 × 14	[1 × 1 conv 3 × 3 conv] × 24	[1 × 1 conv 3 × 3 conv] × 32	[1 × 1 conv 3 × 3 conv] × 48	[1 × 1 conv 3 × 3 conv] × 64
Transition Layer (3)	14 × 14	1 × 1 conv			
	7 × 7	2 × 2 average pool, stride 2			
Dense Block (4)	7 × 7	[1 × 1 conv 3 × 3 conv] × 16	[1 × 1 conv 3 × 3 conv] × 32	[1 × 1 conv 3 × 3 conv] × 32	[1 × 1 conv 3 × 3 conv] × 48
Classification Layer	1 × 1	7 × 7 global average pool			
		1000D fully-connected, softmax			

(a) Transfer module in DenseNet

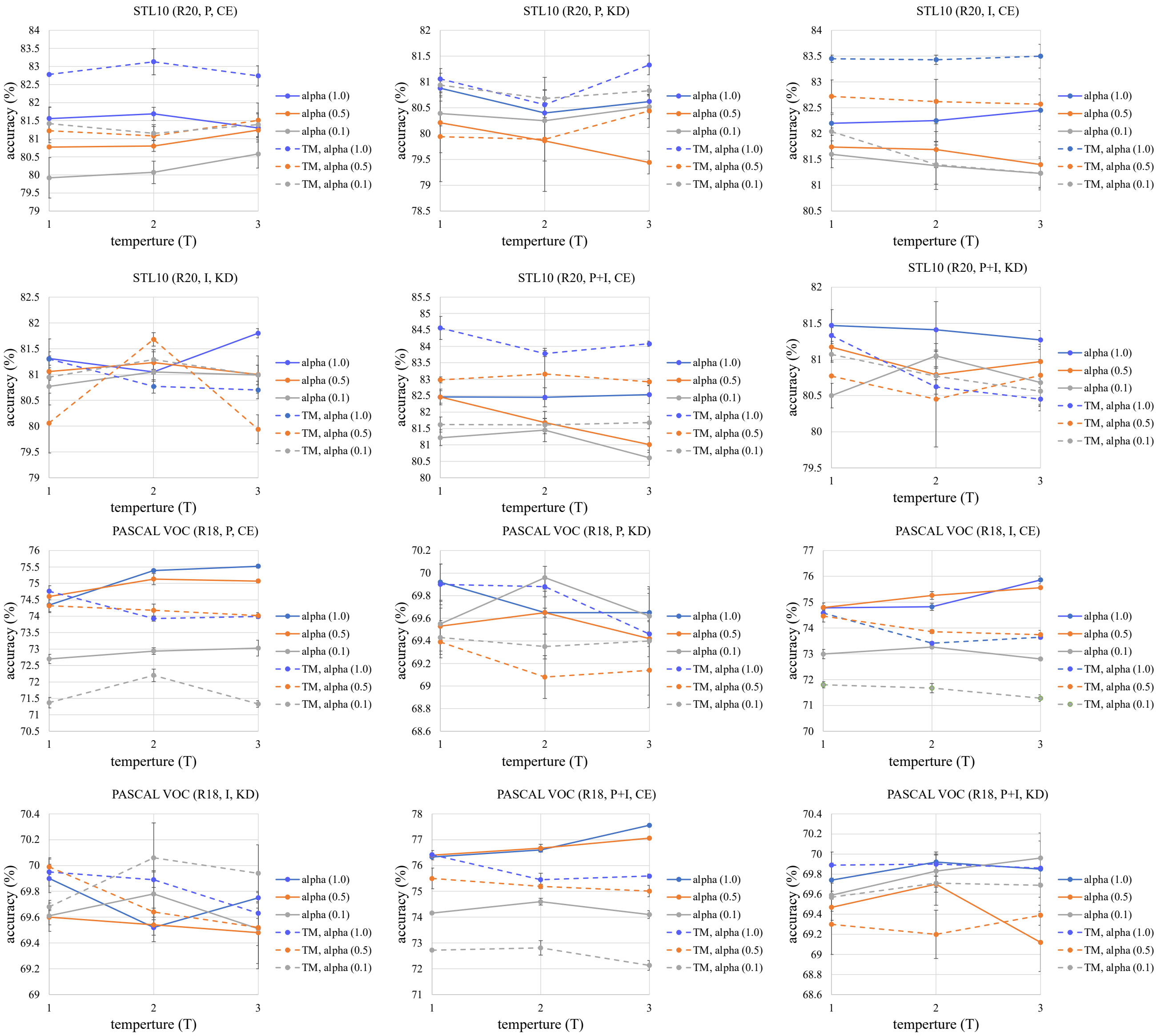
Model	Block	conv1	conv2_x		conv3_x		conv4_x		conv5_x		
			F	N	F	N	F	N	F	N	
ResNet-{18, 34}	Basic	conv, 7 × 7 × 7, 64, temporal stride 1, spatial stride 2	64	{2, 3}	128	{2, 4}	256	{2, 6}	512	{2, 3}	global average pool, C-d fully-connected, softmax
ResNet-{50, 101, 152, 200}	Bottleneck		64	3	128	{4, 4, 8, 24}	256	{6, 23, 36, 36}	512	3	
Pre-act ResNet-200	Pre-act		64	3	128	24	256	36	512	3	
WRN-50	Bottleneck		128	3	256	4	512	6	1024	3	
ResNeXt-101	ResNeXt		128	3	256	24	512	36	1024	3	
DenseNet-{121, 201}	DenseNet		64	{6, 6}	128	{12, 12}	256	{24, 48}	{512, 896}	{16, 32}	

(b) Transfer module in 3D-ResNet

Figure 1. **Schematic of the transfer modules for efficient SSKT with different CNN architectures.** The transfer module used in the SSKT consists of summation of feature output of bottleneck layers from each convolutional block. Schematic shows and example of the transfer module with different CNN architectures for SSKT using multiple sources.

[9] S. Liu, A. J. Davison, and E. Johns. Self-supervised generalisation with meta auxiliary learning. In *In Proc. of NeurIPS*, 2019. 2, 8

[10] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C.



STL10 (R20, P, CE)

STL10 (R20, P, KD)

STL10 (R20, I, CE)

STL10 (R20, I, KD)

STL10 (R20, P+I, CE)

STL10 (R20, P+I, KD)

PASCAL VOC (R18, P, CE)

PASCAL VOC (R18, P, KD)

PASCAL VOC (R18, I, CE)

PASCAL VOC (R18, I, KD)

PASCAL VOC (R18, P+I, CE)

PASCAL VOC (R18, P+I, KD)

Figure 2. **Parameter optimization of SSKT.** The title of each graph is composed of D_t (target model, T_s, auxiliary loss). T is the temparture parameter of each auxiliary loss, and α is the balance parameter of the total loss.

Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *In Proc. of CVPR*, 2018. 1

[11] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *In Proc. of ICLR*, 2015. 8

[12] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. In *CoRR abs/1212.0402*, 2012. 2

[13] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1452–1464, 2018. 1

Table 2. Performance change according to the configuration of the SSKT for CIFAR100 dataset compared to the training from scratch.

T_s	T_t	Model	Method	TM	Loss	acc.
-	C100	R20	scratch	-	CE	68.26±0.36
P		R20	SSKT	x	CE+CE	67.65±0.21
		R20	SSKT	x	CE+KD	68.01±0.42
		R20	SSKT	o	CE+CE	67.96±0.27
		R20	SSKT	o	CE+KD	67.83±0.34
I		R20	SSKT	x	CE+CE	68.3±0.17
		R20	SSKT	x	CE+KD	68.37±0.23
		R20	SSKT	o	CE+CE	**68.63±0.12**
		R20	SSKT	o	CE+KD	68.35±0.1
P+I		R20	SSKT	x	CE+CE	67.87±0.17
		R20	SSKT	x	CE+KD	68.13±0.05
		R20	SSKT	o	CE+CE	68.56±0.23
		R20	SSKT	o	CE+KD	67.84±0.28
-		R32	scratch	-	CE	70.33±0.19
P		R32	SSKT	x	CE+CE	69.97±0.16
		R32	SSKT	x	CE+KD	69.93±0.21
		R32	SSKT	o	CE+CE	69.69±0.19
		R32	SSKT	o	CE+KD	69.92±0.31
I		R32	SSKT	x	CE+CE	70.6±0.05
		R32	SSKT	x	CE+KD	70.17±0.14
		R32	SSKT	o	CE+CE	70.75±0.06
		R32	SSKT	o	CE+KD	70.0±0.11
P+I		R32	SSKT	x	CE+CE	69.25±0.58
		R32	SSKT	x	CE+KD	69.22±0.43
		R32	SSKT	o	CE+CE	**70.94±0.36**
		R32	SSKT	o	CE+KD	69.44±0.01

Table 3. Performance change according to the configuration of the SSKT for STL10 dataset compared to the training from scratch.

T_s	T_t	Model	Method	TM	Loss	acc.
-	STL10	R20	scratch	-	CE	81.15±0.34
P		R20	SSKT	x	CE+CE	81.56±0.32
		R20	SSKT	x	CE+KD	80.88±0.19
		R20	SSKT	o	CE+CE	82.76±0.05
		R20	SSKT	o	CE+KD	81.06±0.2
I		R20	SSKT	x	CE+CE	82.2±0.17
		R20	SSKT	x	CE+KD	80.82±0.14
		R20	SSKT	o	CE+CE	83.45±0.07
		R20	SSKT	o	CE+KD	81.3±0.39
P+I		R20	SSKT	x	CE+CE	82.46±0.24
		R20	SSKT	x	CE+KD	81.47±0.22
		R20	SSKT	o	CE+CE	**84.56±0.35**
		R20	SSKT	o	CE+KD	81.33±0.11
-		R32	scratch	-	CE	81.19±0.17
P		R32	SSKT	x	CE+CE	82.1±0.14
		R32	SSKT	x	CE+KD	81.29±0.22
		R32	SSKT	o	CE+CE	83.06±0.27
		R32	SSKT	o	CE+KD	81.19±0.12
I		R32	SSKT	x	CE+CE	82.88±0.33
		R32	SSKT	x	CE+KD	81.4±0.23
		R32	SSKT	o	CE+CE	83.68±0.28
		R32	SSKT	o	CE+KD	81.76±0.18
P+I		R32	SSKT	x	CE+CE	82.39±0.15
		R32	SSKT	x	CE+KD	79.8±0.47
		R32	SSKT	o	CE+CE	**83.4±0.2**
		R32	SSKT	o	CE+KD	80.05±1.06

Table 4. Performance change according to the configuration of the SSKT for ImageNet and Places365 compared to the training from scratch.

T_s	T_t	Model	Method	TM	Loss	acc.
-	P	R18	scratch	-	CE	50.92
P		R18	SSKT	x	CE+CE	54.41
		R18	SSKT	x	CE+KD	53.42
		R18	SSKT	o	CE+CE	54.5
		R18	SSKT	o	CE+KD	54.11
I		R18	SSKT	x	CE+CE	53.47
		R18	SSKT	x	CE+KD	53.51
		R18	SSKT	o	CE+CE	53.67
		R18	SSKT	o	CE+KD	53.44
P+I		R18	SSKT	x	CE+CE	**54.78**
		R18	SSKT	x	CE+KD	54.5
		R18	SSKT	o	CE+CE	54.62
		R18	SSKT	o	CE+KD	54.5
-	I	R18	scratch	-	CE	64.14
P		R18	SSKT	x	CE+CE	64.18
		R18	SSKT	x	CE+KD	64.21
		R18	SSKT	o	CE+CE	64.99
		R18	SSKT	o	CE+KD	63.53
I		R18	SSKT	x	CE+CE	67.79
		R18	SSKT	x	CE+KD	66.0
		R18	SSKT	o	CE+CE	67.46
		R18	SSKT	o	CE+KD	65.65
P+I		R18	SSKT	x	CE+CE	**70.57**
		R18	SSKT	x	CE+KD	67.42
		R18	SSKT	o	CE+CE	67.64
		R18	SSKT	o	CE+KD	66.81

Table 5. Performance change according to the configuration of the SSKT for PASCAL VOC compared to the training from scratch.

T_s	T_t	Model	Method	TM	Loss	acc.
-	VOC	R18	scratch	-	BCE	67.28±0.25
P		R18	SSKT	x	BCE+CE	74.34±0.23
		R18	SSKT	x	BCE+KD	69.92±0.16
		R18	SSKT	o	BCE+CE	74.76±0.17
		R18	SSKT	o	BCE+KD	69.9±0.18
I		R18	SSKT	x	BCE+CE	74.78±0.09
		R18	SSKT	x	BCE+KD	69.9±0.35
		R18	SSKT	o	BCE+CE	74.58±0.11
		R18	SSKT	o	BCE+KD	69.95±0.19
P+I		R18	SSKT	x	BCE+CE	76.33±0.0
		R18	SSKT	x	BCE+KD	69.74±0.17
		R18	SSKT	o	BCE+CE	**76.42**±**0.06**
		R18	SSKT	o	BCE+KD	69.89±0.13
-		R34	scratch	-	BCE	66.0±0.49
P		R34	SSKT	x	BCE+CE	73.83±0.38
		R34	SSKT	x	BCE+KD	69.93±0.03
		R34	SSKT	o	BCE+CE	75.65±0.12
		R34	SSKT	o	BCE+KD	69.51±0.13
I		R34	SSKT	x	BCE+CE	74.25±0.12
		R34	SSKT	x	BCE+KD	70.05±0.14
		R34	SSKT	o	BCE+CE	75.14±0.14
		R34	SSKT	o	BCE+KD	70.18±0.11
P+I		R34	SSKT	x	BCE+CE	75.88±0.1
		R34	SSKT	x	BCE+KD	70.15±0.09
		R34	SSKT	o	BCE+CE	**77.02**±**0.02**
		R34	SSKT	o	BCE+KD	70.58±0.35
-		R50	scratch	-	BCE	61.16±0.34
P		R50	SSKT	x	BCE+CE	63.29±1.43
		R50	SSKT	x	BCE+KD	65.5±0.2
		R50	SSKT	o	BCE+CE	74.44±0.06
		R50	SSKT	o	BCE+KD	65.94±0.09
I		R50	SSKT	x	BCE+CE	63.96±2.74
		R50	SSKT	x	BCE+KD	66.11±0.32
		R50	SSKT	o	BCE+CE	74.24±0.05
		R50	SSKT	o	BCE+KD	65.77±0.13
P+I		R50	SSKT	x	BCE+CE	69.27±0.21
		R50	SSKT	x	BCE+KD	66.0±0.29
		R50	SSKT	o	BCE+CE	**77.1**±**0.14**
		R50	SSKT	o	BCE+KD	66.22±0.23

Table 6. Performance change according to the configuration of the SSKT for UCF101 and HMDB51 compared to the training from scratch.

T_s	T_t	Model	Method	TM	Loss	acc.
-	U101	3DR18	scratch	-	CE	43.28
P		3DR18	SSKT	x	CE+CE	44.1
		3DR18	SSKT	x	CE+KD	44.79
		3DR18	SSKT	o	CE+CE	45.35
		3DR18	SSKT	o	CE+KD	43.95
I		3DR18	SSKT	x	CE+CE	46.62
		3DR18	SSKT	x	CE+KD	40.35
		3DR18	SSKT	o	CE+CE	44.26
		3DR18	SSKT	o	CE+KD	38.95
P+I		3DR18	SSKT	x	CE+CE	**52.19**
		3DR18	SSKT	x	CE+KD	43.68
		3DR18	SSKT	o	CE+CE	47.09
		3DR18	SSKT	o	CE+KD	45.0
-	H51	3DR18	scratch	-	CE	17.14
P		3DR18	SSKT	x	CE+CE	18.18
		3DR18	SSKT	x	CE+KD	17.33
		3DR18	SSKT	o	CE+CE	18.77
		3DR18	SSKT	o	CE+KD	17.59
I		3DR18	SSKT	x	CE+CE	18.64
		3DR18	SSKT	x	CE+KD	18.12
		3DR18	SSKT	o	CE+CE	18.38
		3DR18	SSKT	o	CE+KD	18.77
P+I		3DR18	SSKT	x	CE+CE	19.75
		3DR18	SSKT	x	CE+KD	18.38
		3DR18	SSKT	o	CE+CE	**20.54**
		3DR18	SSKT	o	CE+KD	17.99

Table 7. Performance change according to the configuration of the SSKT for STL10 compared to the training from scratch with MobileNet V2 (MV2) and DenseNet121 (D121).

T_s	T_t	Model	Method	TM	Loss	acc.
-	S10	MV2	scratch	-	CE	72.26±0.83
P		MV2	SSKT	x	CE+CE	75.79±0.19
		MV2	SSKT	x	CE+KD	74.0±0.35
		MV2	SSKT	o	CE+CE	75.28±0.49
		MV2	SSKT	o	CE+KD	73.37±1.8
I		MV2	SSKT	x	CE+CE	76.08±0.63
		MV2	SSKT	x	CE+KD	74.39±0.82
		MV2	SSKT	o	CE+CE	75.35±0.61
		MV2	SSKT	o	CE+KD	72.6±0.67
P+I		MV2	SSKT	x	CE+CE	76.69±0.18
		MV2	SSKT	x	CE+KD	73.29±0.89
		MV2	SSKT	o	CE+CE	**76.96±0.39**
		MV2	SSKT	o	CE+KD	73.35±0.99
-		D121	scratch	-	CE	72.02±0.48
P		D121	SSKT	x	CE+CE	76.17±0.35
		D121	SSKT	x	CE+KD	74.83±0.59
		D121	SSKT	o	CE+CE	73.46±0.62
		D121	SSKT	o	CE+KD	72.55±0.43
I		D121	SSKT	x	CE+CE	76.0±0.33
		D121	SSKT	x	CE+KD	73.7±0.22
		D121	SSKT	o	CE+CE	74.35±0.3
		D121	SSKT	o	CE+KD	71.13±0.59
P+I		D121	SSKT	x	CE+CE	**77.03±0.17**
		D121	SSKT	x	CE+KD	73.76±0.84
		D121	SSKT	o	CE+CE	76.09±0.26
		D121	SSKT	o	CE+KD	70.94±1.14

Table 8. SSKT results for PASCAL VOC, UCF101, and HMDB51 using pretrained weights. *ft* stands for finetuning and K stands for Kinetics-400 dataset (Kay et al. 2017).

T_s	T_t	Model	Method	TM	Loss	acc.
-	VOC	R18	*ft* (I)	-	CE	90.52±0.11
P		R18	SSKT	x	CE+CE	89.3±0.04
		R18	SSKT	x	CE+KD	**92.28±0.06**
		R18	SSKT	o	CE+CE	90.83±0.04
		R18	SSKT	o	CE+KD	92.21±0.05
I		R18	SSKT	x	CE+CE	91.29±0.03
		R18	SSKT	x	CE+KD	92.26±0.07
		R18	SSKT	o	CE+CE	91.58±0.15
		R18	SSKT	o	CE+KD	92.19±0.09
P+I		R18	SSKT	x	CE+CE	91.28±0.05
		R18	SSKT	x	CE+KD	92.19±0.07
		R18	SSKT	o	CE+CE	91.25±0.08
		R18	SSKT	o	CE+KD	92.25±0.07
-	U101	3DR18	*ft* (K)	-	CE	83.95
P		3DR18	SSKT	x	CE+CE	84.53
		3DR18	SSKT	x	CE+KD	**84.58**
		3DR18	SSKT	o	CE+CE	83.87
		3DR18	SSKT	o	CE+KD	83.98
I		3DR18	SSKT	x	CE+CE	81.99
		3DR18	SSKT	x	CE+KD	83.42
		3DR18	SSKT	o	CE+CE	84.29
		3DR18	SSKT	o	CE+KD	84.37
P+I		3DR18	SSKT	x	CE+CE	78.56
		3DR18	SSKT	x	CE+KD	84.14
		3DR18	SSKT	o	CE+CE	82.81
		3DR18	SSKT	o	CE+KD	84.19
-	H51	3DR18	*ft* (K)	-	CE	56.64
P		3DR18	SSKT	x	CE+CE	56.77
		3DR18	SSKT	x	CE+KD	56.77
		3DR18	SSKT	o	CE+CE	57.75
		3DR18	SSKT	o	CE+KD	**57.82**
I		3DR18	SSKT	x	CE+CE	56.18
		3DR18	SSKT	x	CE+KD	56.9
		3DR18	SSKT	o	CE+CE	53.3
		3DR18	SSKT	o	CE+KD	57.75
P+I		3DR18	SSKT	x	CE+CE	54.48
		3DR18	SSKT	x	CE+KD	56.05
		3DR18	SSKT	o	CE+CE	57.29
		3DR18	SSKT	o	CE+KD	57.1

Table 9. Comparison with MAXL [9] according to the configuration of the SSKT. V16 denotes VGG16 [11]. F denotes focal loss [9]. C denotes cross-entropy loss. TM denotes transfer module.

T_t	T_t Model	MAXL ($\psi[i]$)	SSKT (T_s, TM)	T_s Model
C10	V16 (F)	93.27±0.09 (2)	93.56±0.02 (I, x)	V16
	V16 (F)	93.47±0.08 (3)	94.1±0.1 (I, o)	V16
	V16 (F)	93.49±0.05 (5)	92.94±0.02 (I, x)	R50
	V16 (F)	93.10±0.08 (10)	94.1±0.1 (I, o)	R50
	V16 (F)	-	92.56±0.15 (P, x)	R50
	V16 (F)	-	92.72±0.15 (P, o)	R50
	V16 (F)	-	92.6±0.19 (P+I, x)	R50, V16
	V16 (F)	-	93.06±0.2 (P+I, o)	R50, V16
	V16 (F)	-	92.2±0.08 (P+I, x)	R50, R50
	V16 (F)	-	92.54±0.17 (P+I, o)	R50, R50
	V16 (C)	-	93.78±0.04 (I, x)	V16
	V16 (C)	-	**94.22±0.02** (I, o)	V16
	V16 (C)	-	93.08±0.06 (I, x)	R50
	V16 (C)	-	93.78±0.04 (I, o)	R50
	V16 (C)	-	93.04±0.15 (P, x)	R50
	V16 (C)	-	93.12±0.12 (P, o)	R50
	V16 (C)	-	93.35±0.21 (P+I, x)	R50, V16
	V16 (C)	-	93.67±0.17 (P+I, o)	R50, V16
	V16 (C)	-	93.02±0.11 (P+I, x)	R50, R50
	V16 (C)	-	93.26±0.12 (P+I, o)	R50, R50
	R20 (F)	91.53±0.33 (2)	90.52±0.34 (I, x)	V16
	R20 (F)	91.52±0.1 (3)	91.48±0.03 (I, o)	V16
	R20 (F)	91.38±0.47 (5)	90.88±0.02 (I, x)	R50
	R20 (F)	91.56±0.16 (10)	91.66±0.09 (I, o)	R50
	R20 (F)	-	89.97±0.02 (P, x)	R50
	R20 (F)	-	91.26±0.17 (P, o)	R50
	R20 (F)	-	89.42±0.08 (P+I, x)	R50, V16
	R20 (F)	-	90.93±0.01 (P+I, o)	R50, V16
	R20 (F)	-	90.02±0.21 (P+I, x)	R50, R50
	R20 (F)	-	91.11±0.18 (P+I, o)	R50, R50
	R20 (C)	-	92.28±0.07 (I, x)	R50
	R20 (C)	-	92.44±0.05 (I, o)	R50
	R20 (C)	-	92.21±0.06 (P, x)	R50
	R20 (C)	-	92.25±00.04 (P, o)	R50
	R20 (C)	-	91.9±0.1 (P+I, x)	R50, R50
	R20 (C)	-	**92.46±0.15** (P+I, o)	R50, R50

.