6 Supplementary Material: Learning to Project for Cross-Task Knowledge Distillation (Submission 448)

6.1 Code

We include our anonymised code in the accompanying .zip file, and will release it publicly upon acceptance. Our codebase is written in Python using PyTorch [56] and is based on several other codebases:

- **Monocular depth estimation student:** we build on the codebase of AiT [54], which is built in turn on MMDetection [53]. No license is specified in the original AiT repository. MMDetection is released under the Apache license.
- **Semantic segmentation:** our code builds on the pipeline of `github.com/yassouali/pytorch-segmentation` (released under MIT license).
- **Image-to-image translation:** our code builds on the official PyTorch implementation of Pix2Pix and CycleGAN (released under the BSD license): `github.com/junyanz/pytorch-CycleGAN-and-pix2pix`

A full README is included in the code release, and includes instructions for setup, evaluation, and training.

6.2 Full monocular depth estimation results

Table 1 shows only some metrics for monocular depth estimation (MDE) on NYUv2 [68] for the sake of brevity. We include here four additional tables showing the performance on all available metrics of a depth estimation student when distilled to from teachers trained for different tasks. Each table also includes metrics for the baseline, which is a student model trained without any distillation setup of any kind (i.e. only the task loss $\mathcal{L}_{task}$ is used). Full details of the metrics used are in section 6.5. See section 6.3.1 for complete architectural details and section 6.4 for loss function details.

Each table shows a single teacher/student task pair, and compares four different knowledge distillation methods when using both the traditional projection and our inverted projection, as well as including a percentage *Improvement* showing the difference in performance when using our inverted projection compared to the traditional projection.

Table 4 shows results using a depth estimation teacher. As the teacher and student tasks are identical and the task-specific features in the teacher are desired for the student model, the traditional projection produces a greater performance improvement than our inverted projection.

Table 5 shows results using an instance segmentation teacher. Instance segmentation produces both semantic labels and instance labels, and the semantic masks and labels are known to be useful for depth estimation (see section 2), so while the teacher and student tasks are different, they are similar to one another. Therefore, we see that the traditional projection still outperforms our inverted projection in most cases, as expected.

Table 6 shows results with a classification teacher. This is a cross-task setup: classification is relatively unrelated to monocular depth estimation. As a result, it can be seen that our inverted projection outperforms the traditional projection.

		🟨 **Depth** ***(Most similar)*** ⟶ Depth						
Method	Projection type	$\delta_1 \uparrow$	$\delta_2 \uparrow$	$\delta_3 \uparrow$	Abs. Rel. ↓	Sq. Rel. ↓	RMS ↓	RMSL ↓
No teacher (baseline)		*0.845* ±0.007	*0.974* ±0.001	*0.995* ±0.000	*0.127* ±0.003	*0.078* ±0.002	*0.440* ±0.005	*0.160* ±0.003
FitNets [54] *ICLR 2015*	Traditional	**0.868**	**0.979**	**0.996**	**0.117**	**0.069**	**0.406**	**0.148**
	Inverted (Ours)	0.849	0.976	0.995	0.124	0.075	0.432	0.157
	Improvement	*-2.17%*	*-0.34%*	*-0.01%*	*-6.35%*	*-9.66%*	*-6.49%*	*-5.92%*
AT [82] *ICLR 2017*	Traditional	**0.856**	0.976	**0.995**	**0.122**	**0.073**	0.426	**0.155**
	Inverted (Ours)	**0.856**	**0.977**	**0.995**	**0.122**	**0.073**	**0.425**	**0.155**
	Improvement	*-0.11%*	*0.06%*	*0.01%*	*-0.08%*	*0.82%*	*0.02%*	*0.00%*
PKT [55] *ECCV 2018*	Traditional	**0.854**	**0.978**	**0.996**	**0.122**	**0.072**	0.429	**0.155**
	Inverted (Ours)	**0.854**	0.977	**0.996**	**0.122**	0.073	**0.427**	**0.155**
	Improvement	*0.04%*	*-0.09%*	*0.02%*	*-0.16%*	*-1.38%*	*0.42%*	*0.19%*
Ensemble [15] *NeurIPS 2022*	Traditional	**0.861**	**0.978**	**0.996**	**0.119**	**0.070**	**0.416**	**0.151**
	Inverted (Ours)	0.849	0.975	**0.996**	0.124	0.076	0.433	0.157
	Improvement	*-1.46%*	*-0.29%*	*-0.05%*	*-4.64%*	*-7.86%*	*-4.11%*	*-4.25%*

Table 4: **Depth teacher → Depth student (no task gap).** As expected, in same-task settings, our inverted projection produces a smaller improvement than the traditional projection. *Improvement* is % change using our inverted projection over using the traditional projection. See section 6.3 for model details. Baseline ± figures are variance from 3 runs.

		🟩 **Instance Segmentation** ⟶ Depth						
Method	Projection type	$\delta_1 \uparrow$	$\delta_2 \uparrow$	$\delta_3 \uparrow$	Abs. Rel. ↓	Sq. Rel. ↓	RMS ↓	RMSL ↓
No teacher (baseline)		*0.845* ±0.007	*0.974* ±0.001	*0.995* ±0.000	*0.127* ±0.003	*0.078* ±0.002	*0.440* ±0.005	*0.160* ±0.003
FitNets [54] *ICLR 2015*	Traditional	**0.855**	**0.977**	**0.996**	**0.122**	**0.073**	**0.425**	**0.154**
	Inverted (Ours)	0.851	0.975	0.995	0.124	0.075	0.431	0.157
	Improvement	*-0.41%*	*-0.25%*	*-0.02%*	*-1.78%*	*-2.52%*	*-1.31%*	*-1.68%*
AT [82] *ICLR 2017*	Traditional	0.852	0.976	**0.995**	0.123	0.075	0.431	0.156
	Inverted (Ours)	**0.855**	**0.978**	**0.995**	**0.121**	**0.073**	**0.429**	**0.155**
	Improvement	*0.42%*	*0.16%*	*0.00%*	*1.38%*	*1.74%*	*0.53%*	*0.77%*
PKT [55] *ECCV 2018*	Traditional	**0.857**	**0.976**	**0.995**	**0.123**	**0.075**	**0.427**	**0.155**
	Inverted (Ours)	0.854	**0.976**	**0.995**	**0.123**	**0.075**	0.429	0.156
	Improvement	*-0.34%*	*0.04%*	*0.01%*	*-0.08%*	*-0.13%*	*-0.44%*	*-0.52%*
Ensemble [15] *NeurIPS 2022*	Traditional	**0.856**	**0.977**	**0.996**	**0.122**	**0.072**	**0.425**	**0.154**
	Inverted (Ours)	0.848	0.975	0.995	0.124	0.076	0.435	0.157
	Improvement	*-0.95%*	*-0.16%*	*-0.05%*	*-1.64%*	*-4.70%*	*-2.16%*	*-1.88%*

Table 5: **Instance segmentation teacher → Depth student (small task gap).** The two tasks are different, but are similar enough that the traditional projection produces greater improvements than our inverted projection does with most methods. *Improvement* is % change using our inverted projection over using the traditional projection. See section 6.3 for model details. Baseline ± figures are variance from 3 runs.

		■ **Classification** ⟶ Depth						
Method	Projection type	$\delta_1 \uparrow$	$\delta_2 \uparrow$	$\delta_3 \uparrow$	Abs. Rel. ↓	Sq. Rel. ↓	RMS ↓	RMSL ↓
No teacher (baseline)		*0.845* ±0.007	*0.974* ±0.001	*0.995* ±0.000	*0.127* ±0.003	*0.078* ±0.002	*0.440* ±0.005	*0.160* ±0.003
FitNets [54] *ICLR 2015*	Traditional	0.845	**0.976**	**0.996**	0.125	0.076	0.439	0.158
	Inverted (Ours)	**0.850**	0.975	**0.995**	**0.124**	**0.075**	**0.434**	**0.157**
	Improvement	*0.50%*	*-0.06%*	*-0.03%*	*0.53%*	*0.44%*	*1.34%*	*0.80%*
AT [82] *ICLR 2017*	Traditional	0.850	**0.976**	**0.995**	0.125	0.076	0.433	0.157
	Inverted (Ours)	**0.853**	**0.976**	**0.995**	**0.123**	**0.074**	**0.430**	**0.156**
	Improvement	*0.35%*	*0.08%*	*0.02%*	*1.61%*	*3.03%*	*0.79%*	*0.83%*
PKT [55] *ECCV 2018*	Traditional	0.851	0.975	**0.996**	0.124	0.076	0.432	0.157
	Inverted (Ours)	**0.853**	**0.976**	0.995	**0.123**	**0.074**	**0.431**	**0.156**
	Improvement	*0.25%*	*0.05%*	*-0.03%*	*1.29%*	*2.50%*	*0.30%*	*0.64%*
Ensemble [15] *NeurIPS 2022*	Traditional	**0.852**	**0.976**	**0.995**	**0.124**	**0.075**	**0.431**	**0.156**
	Inverted (Ours)	0.847	0.975	**0.995**	0.125	0.076	0.437	0.158
	Improvement	*-0.63%*	*-0.12%*	*0.02%*	*-0.89%*	*-1.61%*	*-1.30%*	*-1.09%*

Table 6: **Classification teacher → Depth student (larger task gap).** The two tasks are different enough that the setting becomes more "cross-task" than "same-task", and our inverted projection begins to outperform the traditional student model in terms of improvement over the baseline. *Improvement* is % change using our inverted projection over using the traditional projection. See section 6.3 for model details. Baseline ± figures are variance from 3 runs.

		■ **Random** ***(Least similar)*** ⟶ Depth						
Method	Projection type	$\delta_1 \uparrow$	$\delta_2 \uparrow$	$\delta_3 \uparrow$	Abs. Rel. ↓	Sq. Rel. ↓	RMS ↓	RMSL ↓
No teacher (baseline)		*0.845* ±0.007	*0.974* ±0.001	*0.995* ±0.000	*0.127* ±0.003	*0.078* ±0.002	*0.440* ±0.005	*0.160* ±0.003
FitNets [54] *ICLR 2015*	Traditional	0.828	0.970	**0.995**	0.134	0.084	0.455	0.167
	Inverted (Ours)	**0.851**	**0.976**	**0.995**	**0.124**	**0.075**	**0.431**	**0.156**
	Improvement	*2.86%*	*0.58%*	*0.08%*	*7.47%*	*11.15%*	*5.20%*	*6.36%*
AT [82] *ICLR 2017*	Traditional	**0.857**	**0.977**	**0.996**	**0.121**	**0.073**	**0.428**	**0.154**
	Inverted (Ours)	**0.857**	0.976	0.995	0.122	0.074	**0.428**	0.155
	Improvement	*0.05%*	*-0.04%*	*-0.03%*	*-0.83%*	*-1.37%*	*0.09%*	*-0.39%*
PKT [55] *ECCV 2018*	Traditional	0.856	0.975	**0.995**	0.123	0.075	0.429	**0.155**
	Inverted (Ours)	**0.858**	**0.976**	**0.995**	**0.122**	**0.073**	**0.426**	**0.155**
	Improvement	*0.29%*	*0.10%*	*0.00%*	*1.22%*	*2.80%*	*0.84%*	*0.58%*
Ensemble [15] *NeurIPS 2022*	Traditional	0.835	0.973	0.995	0.128	0.079	0.446	0.162
	Inverted (Ours)	**0.849**	**0.976**	**0.996**	**0.124**	**0.075**	**0.432**	**0.157**
	Improvement	*1.74%*	*0.26%*	*0.06%*	*2.75%*	*4.96%*	*3.03%*	*3.39%*

Table 7: **Randomly-initialised teacher → Depth student (largest task gap).** Our inverted projection produces significant improvement. *Improvement* is % change using our inverted projection over using the traditional projection. See section 6.3 for model details. Baseline ± figures are variance from 3 runs.

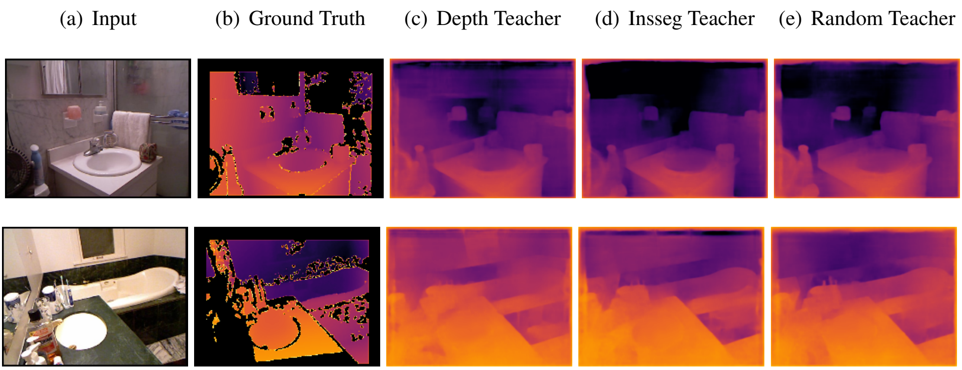

(a) Input (b) Ground Truth (c) Depth Teacher (d) Insseg Teacher (e) Random Teacher

Figure 2: **Qualitative results on NYUv2 (depth) using different teacher tasks:** results from depth estimation, instance segmentation, and randomly-initialised teachers on a MobileNetV2 [24] student. In each case, we use the optimal projection type for the teacher task: the depth teacher's task-specific knowledge is desired, so we use a traditional projection, whereas the instance segmentation and random teachers both have irrelevant knowledge that must be discarded and so perform best with our novel inverted projection, which is able to remove the irrelevant features if needed.

Table 7 shows results with a randomly-initialised and frozen teacher model. The randomly-initialised teacher does not contain any task-specific knowledge whatsoever, and therefore the task-gap between the teacher and the student model is maximised. As this is the most cross-task setting, our inverted projection performs the best in comparison to the traditional projection, as it is only our inverted projection that is able to successfully discard the confounding features present in the randomly-initialised teacher.

We also provide qualitative examples using same-task, similar-task, and randomly-initialised teachers on a different depth estimation student, shown in figure 2. In all cases, we are able to obtain qualitatively good performance.

6.3 Model details

This section details the different student and teacher architectures used for our experiments.

6.3.1 Depth estimation

Teacher models: The teacher models used for depth estimation are:

- **Depth teacher**: SwinV2-B [43] pretrained on NYUv2 as part of the All In Tokens [54] framework. [2], available from the official AiT repository[3].
- **Instance segmentation teacher**: SwinV2-B pretrained on COCO [41] as part of the All In Tokens framework.

[2] https://msravcghub.blob.core.windows.net/ait-release/checkpoint/ait_depth_swinv2b_ar.pth

[3] https://github.com/SwinTransformer/AiT

- **Classification teacher**: ViT-B-16 pretrained on ImageNet-1K, available from the torchvision model hub[4].

Student Models: The depth estimation students used have one of three backbones:

- MobilenetV2 [24]: In our experiments, we use a width multiplier of 0.5.
- EfficientNet-B0 [71].
- ResNet-50 [28].

The decoders used are (names in `this font`):

- `Decoder`: Conv1x1, then 6 blocks of (LeakyReLU + Conv3x3 + LeakyReLU + Conv3x3). At the input to each of the 6 blocks, an incoming skip connection from the encoder is bilinearly upsampled to match the feature resolution, then concatenated to the features.
- `Decoder_dl2`: The same as `Decoder`, except the second Conv3x3 in each block is replaced with a depthwise convolution to reduce parameters.
- `ULightDecoder_skip_4b`: Conv1x1, then 4 blocks of (LeakyReLU + Conv3x3). As in the other decoders, each block receives features from an incoming skip connection, which are upsampled to match the feature resolution and then concatenated.

6.3.2 Semantic segmentation

Teacher models: The teacher models used for semantic segmentation are:

- Segmentation teacher: A DeepLab-V3 [8] with a ResNet-50 backbone, pretrained on a subset of MSCOCO that uses only the 20 categories present in the Pascal VOC dataset. Model and checkpoint loaded from torchvision model hub[4].
- Classification teacher: ResNet-50 [28] pretrained on ImageNet-1K. Model and checkpoint loaded from torchvision model hub[4].

Student model: The student model used for semantic segmentation was a DeepLabV3 [7] with a ResNet50 backbone that is pretrained on ImageNet-1K. The pretrained weights were sourced from the torchvision model hub[4].

6.3.3 Satellite-to-map and Colorization

We use the same teacher models described in section 6.3.1. For the student models on the satellite-to-map experiments, we use a CycleGAN, while for the colorization experiments we use a Pix2Pix model. This Pix2Pix model follows a standard UNet-like architecture with batch norm layers.

[4] https://pytorch.org/vision/stable/models.html

6.4 Task losses

In addition to the projection loss $\mathcal{L}_{distill}$, each student is trained with a task-specific loss $\mathcal{L}_{task}$ to supervise its output. The task and projection loss components are weighted equally, as in equation 2.

The depth task loss function used is a variant of the Scale-Invariant Log-Loss (SILog), first proposed by [19] and modified by [6]:

$$\mathcal{L}_{SILog} = 10\sqrt{\frac{1}{K}\sum_{i=1}^{K} g_i^2 + \frac{0.15}{K^2}\left(\sum_{i=1}^{K} g_i\right)^2} \tag{7}$$

where ground-truth and predicted depth values for pixel i are given as d_i^* and d_i respectively, $g_i = log(d_i) - log(d_i^*)$ and K is the total number of pixels with valid depth values. Semantic segmentation students are trained with a pixelwise cross-entropy loss. For the colourization task we use the vanilla GAN loss[26] in addition to an L1 loss with a weighting of 100.0. For the satellite-to-map translation we use the cyclic consistency loss described in the original CycleGAN paper [86].

6.5 Evaluation metrics

Monocular depth estimation. We use the metrics defined in [19]:

- Abs relative difference (Abs): $\frac{1}{T}\sum_{i=1}^{T}\frac{|d_i - d_i^*|}{d_i^*}$,
- Squared relative difference (Sq): $\frac{1}{T}\sum_{i=1}^{T}\frac{||d_i - d_i^*||^2}{d_i^*}$,
- RMSE (RMS): $\sqrt{\frac{1}{T}\sum_{i=0}^{T}||d_i - d_i^*||^2}$,
- Log RMSE (RMSL): $\sqrt{\frac{1}{T}\sum_{i=0}^{T}||log(d_i) - log(d_i^*)||^2}$,
- The threshold accuracy δ_n: % of d_i s.t. $max(\frac{d_i}{d_i^*}, \frac{d_i^*}{d_i}) = \delta < thr$, where δ_n denotes that $thr = 1.25^n$ (we use $n \in \{1,2,3\}$). T denotes the total number of valid pixels in the ground truth depth map. d_i and d_i^* represent the predicted and ground-truth depth values at pixel i respectively.

6.6 Datasets

For semantic segmentation students, we use the ADE20K Scene Parsing dataset [85], a 150-class subset of the full ADE20K dataset. It contains 20210 training images and 2000 testing images from a variety of indoor and outdoor scenes.

For depth estimation students, the NYUv2 dataset [68] is used, an indoor monocular depth estimation dataset containing 24231 training 654 test examples. Students are trained for 25 epochs.

For image colorization and satellite-to-map translation, we use the CMP Facades [74] and Maps datasets used in Pix2Pix [32], both of which are available from https://efrosgans.eecs.berkeley.edu/pix2pix/datasets/. The Maps dataset was scraped from Google Maps by the Pix2Pix authors.

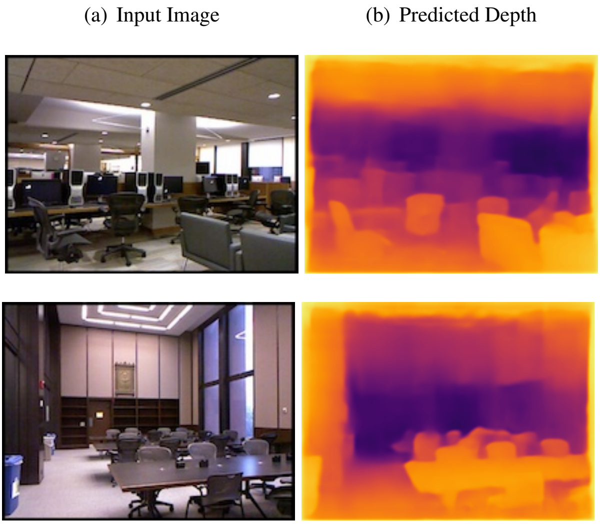


Figure 3: **Qualitative results using a frozen segmentation encoder and frozen depth decoder.** With only a learned linear projection, features from the semantic segmentation task can be made immediately useful for depth.

6.7 Hyperparameters

Monocular depth estimation. Depth estimation students were trained for 25 epochs using the AdamW optimizer with a learning rate of 2e-4 and weight decay of 0.05. The OneCycle learning rate scheduler was used [69] with the maximum learning rate set to 2e-4. The batch size was set to 16.

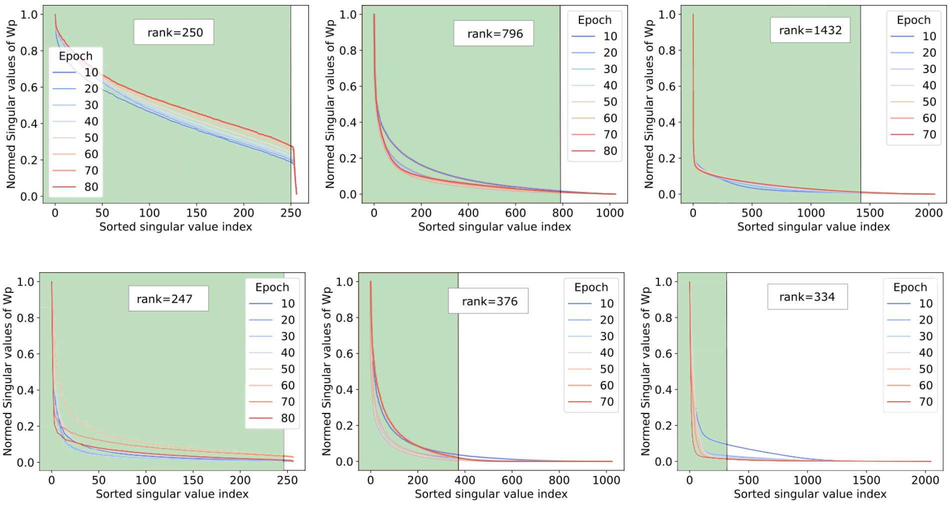
Semantic segmentation. Semantic segmentation students were trained for 80 epochs using the AdamW optimizer with a learning rate of 5e-3 and weight decay of 1e-2. The OneCycle learning rate scheduler was used, with the maximum learning rate set to 5e-3. The batch size was set to 20.

Image-to-image translation (satellite-to-map, colorization). Each model for both of these tasks are trained for 200 epochs using the AdamW optimizer with a learning rate of 2e-4. We keep the initial learning rate for the first 100 epochs and then linearly decay the rate to zero over the next 100 epochs with a batch size of 8.

6.8 Linear mapping between task spaces

In performing cross-task distillation, we assume there is an overlap in information in the representation spaces across different tasks, following both the work in the literature and intuition (see section 2). [47] demonstrated the existence of a learnable linear mapping between text and image features. Concurrently, research has shown that linear projections are very effective for knowledge distillation[15, 49]. However, a natural unification of these two settings has not been explored.

We experimentally verify the validity of this assumption with a simple toy scenario, in



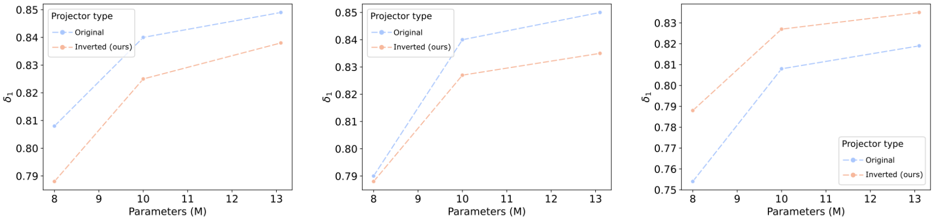
(d) 🟨 Seg. → Seg. (e) 🟦 Class. → Seg. (f) 🟪 Random → Seg.

Figure 4: **Evolution of singular values** of the projection matrix **P** under different cross-task settings and projector types. An L2 loss is used. Green area highlights the rank of **P**. The projection tends towards a higher rank either when using the traditional projection or when using a same-task or similar-task teacher. The low-rank when using our inverted projection in the cross-task setting allows irrelevant features to be filtered out, if necessary for the task pair. **Top row**: traditional projection, **Bottom row**: our inverted projection. Numerical rank is used with a tolerance set to $\sigma_1 * 0.01$.

which a frozen encoder pretrained on instance segmentation is connected via a learnable linear projection to a frozen decoder pretrained for depth estimation. Example qualitative results in figure 3 show that the linearly projected cross-task features can be successfully utilized to generate a coherent output, despite both models being frozen. In fact, by only training the linear projector between these two frozen models, we can attain 0.504 RMSE on NYUv2. This result indicates that a significant portion of the information contained in the instance segmentation features are closely related to the depth estimation task. We conduct additional experiments projecting between various other task representation spaces. These results indicate that cross-task distillation using linear projection is a promising approach for leveraging shared information between specific pairs of tasks, and further motivate our work.

6.9 Training Dynamics of the Inverted Projector

By observing the singular value spectrum of the projector weights and how they evolve over the course of training, we are able to provide further insight into the role of our novel inverted projection for cross-task distillation, as compared to the traditional projection. Figure 4 shows the singular value spectrum of the projector weights throughout the training process of a segmentation student with both similar and different teacher tasks, using either the traditional projection or our novel inverted projection. It can be seen that the traditional projection fails to disregard many of the less-dominant singular values, leading to a higher-rank projection in general. As discussed in section 3.2, this is especially detrimental when there is a significant



(a) Depth → Depth (b) Insseg → Depth (c) Random → Depth

Figure 5: **Comparing performance of different-sized depth students** with both the traditional projection and our novel inverted projection. Where there is knowledge to transfer from teacher to student (i.e. the two tasks are similar), the traditional projection performs better, but when the teacher is random, the opposite is true. Only decoder size is varied. A MobileNetV2 [24] is used as the backbone.

task gap, as it encourages the student to learn task-irrelevant features. When the student model is small, this can significantly degrade the target task performance. However, when using our inverted projection, we observe a consistently lower rank across training for all tasks, compared to the traditional projection. This is because of the inverted projection's ability to suppress the task-irrelevant singular vectors from the teacher model: while the traditional projection remains consistently high-rank regardless of the dissimilarity of the student and teacher tasks, our inverted projection is able to adapt to discard the increasing quantity of undesirable task-specific knowledge encoded in the increasingly dissimilar teacher features.

6.10 Different architecture pairs

To demonstrate the generality of our proposed inverted projection in various cross-task settings, we perform an ablation across several differently-sized student models with similar and dissimilar task pairs. Figure 5 shows that the performance drop or improvement is consistent for both the very small and moderately large student models, across different task pairs. It also mirrors the findings of our previous experiments in sections 4.2, 4.3, and 4.4, showing that the similarity of the teacher and student tasks matters, and that our novel inverted projection performs best when the two tasks are dissimilar.

Table 8 shows results distilling from a classification teacher to a depth student using two student backbones of significantly-different sizes: an EfficientNet-B0 (5.3M params) and a ResNet-50 (25.6M params). These students are chosen to illustrate both a small and a large capacity gap between the student and teacher models. Three different KD methods are used. Our inverted projector outperforms the traditional projector across all metrics for all three KD methods and both student backbone architectures in this cross-task setting, thus demonstrating the generality of our inverted projector for a variety of practical KD settings.

6.11 Teacher-Free Distillation: Results

Table 9 shows the performance of the teacher-free distillation strategy detailed in section 4.5 using different ranks, when applied to a depth estimation setup. A higher value of r uses more of the available principal components to reconstruct the features. The optimal value is found

	Student Arch (backbone) ⟶	EfficientNet-B0 (5.3M)			ResNet-50 (25.6M)		
KD Method	Projection type	$\delta_1 \uparrow$	Abs. ↓	RMS ↓	$\delta_1 \uparrow$	Abs. ↓	RMS ↓
None (baseline)	*N/A*	0.845	0.127	0.440	0.811	0.144	0.480
AT [82]	Original	0.850	0.125	0.433	0.814	**0.143**	0.477
	Inverted (ours)	**0.853**	**0.123**	**0.430**	**0.816**	**0.143**	**0.475**
	Improvement	*0.35%*	*1.61%*	*0.79%*	*0.26%*	*0.14%*	*0.38%*
PKT [55]	Original	0.851	0.124	0.432	0.816	0.143	0.473
	Inverted (ours)	**0.853**	**0.123**	**0.431**	**0.821**	**0.141**	**0.470**
	Improvement	*0.25%*	*1.29%*	*0.30%*	*0.63%*	*1.19%*	*0.70%*
FitNets [64]	Original	0.845	0.125	0.439	0.812	0.145	0.480
	Inverted (ours)	**0.850**	**0.124**	**0.434**	**0.813**	**0.142**	**0.476**
	Improvement	*0.50%*	*0.53%*	*1.34%*	*0.16%*	*1.93%*	*0.77%*

Table 8: **Comparisons with different architecture pairs.** All experiments perform cross-task distillation from a classification teacher to a depth estimation student. The inverted projection is effective across various student model sizes and with different KD methods in this cross-task setting.

Method	RMS ↓	Abs ↓	$\delta_1 \uparrow$
(Baseline) AiT (SwinV2-B) [54]	*0.365*	*0.105*	*0.907*
$\mathcal{L}_{spectral}$ $(r=1)$	0.352	0.105	0.902
$\mathcal{L}_{spectral}$ $(r=2)$	**0.340**	**0.096**	**0.914**
$\mathcal{L}_{spectral}$ $(r=4)$	0.349	0.099	0.911
$\mathcal{L}_{spectral}$ $(r=8)$	0.344	**0.096**	0.910
$\mathcal{L}_{spectral}$ $(r=16)$	0.348	0.098	0.912
$\mathcal{L}_{spectral}$ $(r=32)$	0.347	0.100	0.909

Table 9: **Teacher-free distillation** using our spectral regularisation loss on the NYUv2 dataset using AiT [54] on a SwinV2-b base. The regularisation loss is generally robust to different values of r, with $r=2$ being optimal.

to be when $r=2$.

6.12 Choice of CycleGAN representation

In the encoder-decoder setup, there is a natural choice for the representation to be used as the distillation loss: the representation at the output of the encoder. However, when dealing with different architectures, the decision is less obvious and can significantly impact the efficacy of the distillation process itself. The CycleGAN architecture consists of a discriminator and two separate encoder-decoder models (generators), which we denote here as $G_A(\cdot)$ and $G_B(\cdot)$. The first of these, $G_A(\cdot)$, attempts to generate an image from the input $\mathbf{x}$ that will fool the discriminator, and the second, $G_B(\cdot)$, maps the output of $G_A(\cdot)$ back to the source domain.

Both $G_A(\cdot)$ and $G_B(\cdot)$ are encoder-decoders, and we represent the intermediate features as $G_{A_E}(\cdot)$ and $G_{B_E}(\cdot)$ for each respectively. We trialled the use of each set of features, the results of which are shown in table 10, and found a significant improvement when using the representation from the generator that maps back to the source domain: $\mathbf{Z}_s = G_{B_E}(G_A(\mathbf{x}))$. Therefore, the features from the second generator $G_B(\cdot)$ are those used for feature distillation in our main CycleGAN experiments.

Teacher Task	Position	PSNR ↑	FID ↓
KeyPoint Det.	$\mathbf{Z}_s = G_{B_E}(G_A(x))$	**35.77**	**68.77**
KeyPoint Det.	$\mathbf{Z}_s = G_{A_E}(x)$	34.97	70.28
Image Classif.	$\mathbf{Z}_s = G_{B_E}(G_A(x))$	**36.28**	**59.86**
Image Classif.	$\mathbf{Z}_s = G_{A_E}(x)$	35.94	66.98

Table 10: **Choice of representation** for the distillation loss: either using features from the first generator $G_A(\cdot)$ that generates the synthetic image, or using features from the second generator $G_B(\cdot)$ that maps the synthetic image back to the input domain.

6.13 Analysis of Feature Distillation Loss

This section describes the full analysis of the loss function $\mathcal{L}_{distill}$ detailed in section 3.4 that leads to it breaking into the knowledge transfer component and the regularisation component in equation 5.

6.14 Setup and Definitions

We begin by describing our setup. A teacher model, T, and a student model, S, both take an identical input to produce the teacher and student features, $\mathbf{Z}_t \in \mathbb{R}^{b \times d_t}$ and $\mathbf{Z}_s \in \mathbb{R}^{b \times d_s}$ respectively, where d_t and d_s are the sizes of the feature dimensions for the teacher and student models respectively, and b is the batch size. We also define the inverted projection matrix between the teacher and student feature spaces $\mathbf{P}$. The corresponding projected features would be given by $\bar{\mathbf{Z}}_t = \mathbf{Z}_t\mathbf{P} \in \mathbb{R}^{b \times d_s}$. The rank r for each of these is bounded by:

$$
\begin{aligned}
r_s &= Rank(\mathbf{Z}_s) \leq \min(b, d_s) && (8)\\
r_t &= Rank(\mathbf{Z}_t) \leq \min(b, d_t) && (9)\\
r_p &= Rank(\mathbf{P}) \leq \min(d_t, d_s) && (10)\\
\bar{r}_t &= Rank(\mathbf{Z}_t\mathbf{P}) \leq \min(r_t, r_p) && (11)\\
&\leq \min(\min(b, d_t), \min(d_t, d_s)), && (12)
\end{aligned}
$$

with the latter being due to the fact that $Rank(\mathbf{AB}) \leq \min(Rank(\mathbf{A}), Rank(\mathbf{B}))$.

6.15 Understanding the Inverted Projection

We demonstrate using our inverted projection. The feature distillation loss function is given by:

$$\mathcal{L}_{distill} = \|\bar{\mathbf{Z}}_t - \mathbf{Z}_s\|_2 = \|\mathbf{Z}_t\mathbf{P} - \mathbf{Z}_s\|_2 \tag{13}$$

Taking the singular value decomposition of each of these gives $\bar{\mathbf{Z}}_t = \bar{\mathbf{U}}\bar{\mathbf{\Sigma}}\bar{\mathbf{V}}^T$ and $\mathbf{Z}_s = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$. Using the rank definitions mentioned previously, we can express these as sums of products of the singular values and their corresponding singular vectors, i.e.

$$\bar{\mathbf{Z}}_t = \mathbf{Z}_t\mathbf{P} = \sum_{i=1}^{\bar{r}_t} \bar{\sigma}_i \bar{\mathbf{u}}_i \bar{\mathbf{v}}_i \tag{14}$$

$$\mathbf{Z}_s = \sum_{i=1}^{r_s} \sigma_i \mathbf{u}_i \mathbf{v}_i \tag{15}$$

where $\bar{\boldsymbol{\sigma}} \in \mathbb{R}^{\bar{r}_t}$ and $\boldsymbol{\sigma} \in \mathbb{R}^{r_s}$ denote the columns of $\bar{\boldsymbol{\Sigma}}$ and $\boldsymbol{\Sigma}$ respectively, $\bar{\mathbf{u}} \in \mathbb{R}^{b \times \bar{r}_t}$ and $\mathbf{u} \in \mathbb{R}^{b \times r_s}$ the columns of $\bar{\mathbf{U}}$ and $\mathbf{U}$, and $\bar{\mathbf{v}} \in \mathbb{R}^{\bar{r}_t \times d_s}$, and $\mathbf{v} \in \mathbb{R}^{r_s \times d_s}$ the columns of $\bar{\mathbf{V}}$ and $\mathbf{V}$. The feature distillation loss can be rewritten as:

$$\mathcal{L}_{distill} = \|\bar{\mathbf{Z}}_t - \mathbf{Z}_s\|_2 \tag{16}$$

$$= \underbrace{\|\sum_{i=1}^{\bar{r}_t} \bar{\sigma}_i \bar{\mathbf{u}}_i \bar{\mathbf{v}}_i - \sum_{i=1}^{r_s} \sigma_i \mathbf{u}_i \mathbf{v}_i\|_2}_{\text{knowledge transfer } \textbf{only}} \qquad \textit{1. Same-task} \tag{17}$$

The rank of the inverted projection matrix dictates how the upper bound on the loss can be decomposed into a knowledge transfer component and a regularisation component. We empirically find that **P** works out to be lower-rank in the cross-task setting (see section 6.9), and so $r_s > \bar{r}_t$. This observation allows us to merge the sum in equation 17 such that every one of the $\bar{r}_t$ projected teacher singular values is compared to a student singular value, with the remaining $r_s - \bar{r}_t$ student singular values forming the regularisation term:

$$\mathcal{L}_{distill} = \|\sum_{i=1}^{\bar{r}_t} (\bar{\sigma}_i \bar{\mathbf{u}}_i \bar{\mathbf{v}}_i^T - \sigma_i \mathbf{u}_i \mathbf{v}_i) + \sum_{i=\bar{r}_t+1}^{r_s} \sigma_i \mathbf{u}_i \mathbf{v}_i\|_2$$

$$\leq \underbrace{\|\sum_{i=1}^{\bar{r}_t} \bar{\sigma}_i \bar{\mathbf{u}}_i \bar{\mathbf{v}}_i^T - \sigma_i \mathbf{u}_i \mathbf{v}_i^T\|_2}_{\text{knowledge transfer}} + \underbrace{\|\sum_{i=\bar{r}_t+1}^{r_s} \sigma_i \mathbf{u}_i \mathbf{v}_i^T\|_2}_{\text{student regularisation}} \qquad \textit{2. Cross-task} \tag{18}$$

This shows that, under the cross-task setting, the feature distillation contains both a knowledge transfer component (which incorporates information from the teacher model) and a regularisation component (which acts upon the student model).

References

[1] Jin-Hyun Ahn, Kyungsang Kim, Jeongwan Koh, and Quanzheng Li. Federated active learning (f-al): an efficient annotation strategy for federated learning, 2022.

[2] Dylan Auty and Krystian Mikolajczyk. Monocular Depth Estimation Using Cues Inspired by Biological Vision Systems. In *International Conference on Pattern Recognition (ICPR) 2022*, 2022.

[3] Yucai Bai, Lei Fan, Ziyu Pan, and Long Chen. Monocular Outdoor Semantic Mapping with a Multi-task Network. *arXiv pre-print*, January 2019.

[4] Jason Baldridge and Miles Osborne. Active learning and the total cost of annotation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, 2004.

[5] Kartikeya Bhardwaj, Naveen Suda, and Radu Marculescu. Dream distillation: A data-independent model compression framework. *ICML Joint Workshop on On-Device Machine Learning and Compact Deep Neural Network Representations (ODML-CDNNR)*, 2019.

[6] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. AdaBins: Depth Estimation using Adaptive Bins. *arXiv:2011.14141 [cs]*, November 2020. arXiv: 2011.14141.

[7] Liang-Chieh Chen, George Papandreou, Senior Member, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *TPAMI*, 2017.

[8] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking Atrous Convolution for Semantic Image Segmentation, December 2017. arXiv:1706.05587 [cs].

[9] Liqun Chen, Dong Wang, Zhe Gan, Jingjing Liu, Ricardo Henao, and Lawrence Carin. Wasserstein Contrastive Representation Distillation. *CVPR*, 2020.

[10] Pengguang Chen, Shu Liu, Hengshuang Zhao, and Jiaya Jia. Distilling Knowledge via Knowledge Review. *CVPR*, 2021.

[11] Xianing Chen, Qiong Cao, Yujie Zhong, Jing Zhang, Shenghua Gao, and Dacheng Tao. Dearkd: Data-efficient early knowledge distillation for vision transformers. *CVPR*, 2022.

[12] Xinyang Chen, Sinan Wang, Bo Fu, Mingsheng Long, and Jianmin Wang. Catastrophic forgetting meets negative transfer: Batch spectral shrinkage for safe transfer learning. *NeurIPS*, 2019.

[13] Xinyang Chen, Sinan Wang, Mingsheng Long, and Jianmin Wang. Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation. In *PMLR*, 2019.

[14] Yanbei Chen, Yongqin Xian, A. Sophia Koepke, Ying Shan, and Zeynep Akata. Distilling audio-visual knowledge by compositional contrastive learning. *CVPR*, 2021.

[15] Yudong Chen, Sen Wang, Jiajun Liu, Xuwei Xu, Frank de Hoog, and Zi Huang. Improved Feature Distillation via Projector Ensemble. *NeurIPS*, 2022.

[16] 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.

[17] Zhun Deng, Linjun Zhang, Kailas Vodrahalli, Kenji Kawaguchi, and James Zou. Adversarial Training Helps Transfer Learning via Better Representations. *arXiv preprint*, 6 2021.

[18] 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.

[19] David Eigen, Christian Puhrsch, and Rob Fergus. Depth Map Prediction from a Single Image using a Multi-Scale Deep Network. In *NIPS*, 2014.

[20] Utku Evci, Vincent Dumoulin, Hugo Larochelle, and Michael C Mozer. Head2Toe: Utilizing Intermediate Representations for Better Transfer Learning. In *ICML*. PMLR, 2022.

[21] Gongfan Fang, Jie Song, Xinchao Wang, Chengchao Shen, Xingen Wang, and Mingli Song. Contrastive Model Inversion for Data-Free Knowledge Distillation. *IJCAI*, 2021.

[22] Christopher Fifty, Ehsan Amid, Zhe Zhao, Tianhe Yu, Rohan Anil, and Chelsea Finn. Efficiently identifying task groupings for multi-task learning. *NeurIPS*, 2021.

[23] Michael Fonder and Marc Van Droogenbroeck. Mid-air: A multi-modal dataset for extremely low altitude drone flights. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 0–0, 2019.

[24] Michael H. Fox, Kyungmee Kim, and David Ehrenkrantz. MobileNetV2: Inverted Residuals and Linear Bottlenecks. *CVPR*, 2018.

[25] Ruijiang Gao and Maytal Saar-Tsechansky. Cost-accuracy aware adaptive labeling for active learning. *AAAI*, 2020.

[26] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Networks. *NeurIPS*, 6 2014.

[27] Daniel Greenfeld and Uri Shalit. Robust Learning with the Hilbert-Schmidt Independence Criterion. *arXiv preprint*, 10 2019.

[28] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. ResNet - Deep Residual Learning for Image Recognition. *CVPR*, 2015.

[29] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the Knowledge in a Neural Network. *NeurIPS*, 2015.

[30] Sheng-Jun Huang, Jia-Lve Chen, Xin Mu, and Zhi-Hua Zhou. Cost-effective active learning from diverse labelers. In *IJCAI*, 2017.

[31] Tao Huang, Shan You, Fei Wang, Chen Qian, and Chang Xu. Knowledge distillation from a stronger teacher. *NeurIPS*, 2022.

[32] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-Image Translation with Conditional Adversarial Networks, November 2018. arXiv:1611.07004 [cs].

[33] Stephen James, Paul Wohlhart, Mrinal Kalakrishnan, Dmitry Kalashnikov, Alex Irpan, Julian Ibarz, Sergey Levine, Raia Hadsell, and Konstantinos Bousmalis. Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks. *CVPR*, 2019.

[34] Jianbo Jiao, Ying Cao, Yibing Song, and Rynson Lau. Look Deeper into Depth: Monocular Depth Estimation with Semantic Booster and Attention-Driven Loss. In *ECCV*, 2018.

[35] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *ICCV*, 2023.

[36] Benedikt Kolbeinsson and Krystian Mikolajczyk. DDOS: The Drone Depth and Obstacle Segmentation Dataset. *arXiv preprint arXiv:2312.12494*, 2023.

[37] Benedikt Kolbeinsson and Krystian Mikolajczyk. UCorr: Wire Detection and Depth Estimation for Autonomous Drones. In *International Conference on Robotics, Computer Vision and Intelligent Systems - ROBOVIS*, 2024.

[38] Piotr Komorowski, Hubert Baniecki, and Przemyslaw Biecek. Towards evaluating explanations of vision transformers for medical imaging. In *CVPR Workshops*, June 2023.

[39] Duong H. Le, Trung-Nhan Vo, and Nam Thoa. Paying more Attention to Snapshots of Iterative Pruning : Improving Model Compression via Ensemble Distillation. *BMVC*, 2020.

[40] Deng Li, Aming Wu, Yahong Han, and Qi Tian. Prototype-guided Cross-task Knowledge Distillation for Large-scale Models. *arXiv preprint*, 12 2022.

[41] 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. *ECCV*, 2014.

[42] Yifan Liu, Ke Chen, Chris Liu, Zengchang Qin, Zhenbo Luo, and Jingdong Wang. Structured Knowledge Distillation for Semantic Segmentation. *CVPR*, 2019.

[43] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, Furu Wei, and Baining Guo. Swin Transformer V2: Scaling Up Capacity and Resolution. *CVPR*, 2022.

[44] David Lopez-Paz, Léon Bottou, Bernhard Schölkopf, and Vladimir Vapnik. Unifying distillation and privileged information. *ICLR*, 2016.

[45] Andrey Malinin, Bruno Mlodozeniec, and Mark Gales. Ensemble Distribution Distillation. *ICLR*, 2020.

[46] John Mccormac, Ankur Handa, Stefan Leutenegger, and Andrew J Davison. SceneNet RGB-D: Can 5M Synthetic Images Beat Generic ImageNet Pre-training on Indoor Segmentation? In *ICCV*, 2017.

[47] Jack Merullo, Louis Castricato, Carsten Eickhoff, and Ellie Pavlick. Linearly Mapping from Image to Text Space. *ICLR*, 2023.

[48] Roy Miles and Krystian Mikolajczyk. Cascaded channel pruning using hierarchical self-distillation. *BMVC*, 2020.

[49] Roy Miles and Krystian Mikolajczyk. Understanding the role of the projector in knowledge distillation. *AAAI*, 2024.

[50] Roy Miles, Adrian Lopez Rodriguez, and Krystian Mikolajczyk. Information Theoretic Representation Distillation. *BMVC*, 12 2022.

[51] Roy Miles, Mehmet Kerim Yucel, Bruno Manganelli, and Albert Saa-Garriga. MobileVOS: Real-Time Video Object Segmentation Contrastive Learning meets Knowledge Distillation. *CVPR*, 3 2023.

[52] Roy Miles, Ismail Elezi, and Jiankang Deng. V$_k$d: Improving knowledge distillation using orthogonal projections. *CVPR*, 2024.

[53] MMDetection Contributors. OpenMMLab Detection Toolbox and Benchmark, August 2018. URL https://github.com/open-mmlab/mmdetection. original-date: 2018-08-22T07:06:06Z.

[54] Jia Ning, Chen Li, Zheng Zhang, Zigang Geng, Qi Dai, Kun He, and Han Hu. All in Tokens: Unifying Output Space of Visual Tasks via Soft Token, January 2023. arXiv:2301.02229 [cs].

[55] Nikolaos Passalis and Anastasios Tefas. Learning Deep Representations with Probabilistic Knowledge Transfer. *ECCV*, 2018.

[56] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*, 2019.

[57] Christoph Raab, Philipp Väth, Peter Meier, and Frank-Michael Schleif. Bridging Adversarial and Statistical Domain Transfer via Spectral Adaptation Networks. In *ACCV 2020*. Springer International Publishing, 2020.

[58] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *PMLR*, 2021.

[59] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing Network Design Spaces. *CVPR*, 3 2020.

[60] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *JMLR*, 2020.

[61] Michaël Ramamonjisoa and Vincent Lepetit. SharpNet: Fast and Accurate Recovery of Occluding Contours in Monocular Depth Estimation. *arXiv preprint*, 2019. arXiv: 1905.08598v1.

[62] Sucheng Ren, Zhengqi Gao, Tianyu Hua, Zihui Xue, Yonglong Tian, Shengfeng He, and Hang Zhao. Co-advise: Cross Inductive Bias Distillation. *CVPR*, 2022.

[63] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. FitNets: Hints For Thin Deep Nets. *ICLR*, 2015.

[64] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. FitNets: Hints for Thin Deep Nets. *arXiv preprint*, March 2015. arXiv: 1412.6550.

[65] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *MICCAI*, 2015.

[66] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *NeurIPS Workshop on Energy Efficient Machine Learning and Cognitive Computing*, 2019.

[67] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *NeurIPS Workshop on Energy Efficient Machine Learning and Cognitive Computing*, 2019.

[68] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor Segmentation and Support Inference from RGBD Images. In *ECCV*, 2012.

[69] Leslie N. Smith and Nicholay Topin. Super-Convergence: Very Fast Training of Neural Networks Using Large Learning Rates, May 2018. arXiv:1708.07120 [cs, stat].

[70] Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. VL-ADAPTER: Parameter-Efficient Transfer Learning for Vision-and-Language Tasks. In *CVPR*, 2022.

[71] Mingxing Tan and Quoc V. Le. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In *Proceedings of the 36th International Conference on Machine Learning*, 2019. arXiv: 1905.11946.

[72] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. *ICLR*, 2019.

[73] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *PMLR*, 2021.

[74] Radim Tyleček and Radim Šára. Spatial Pattern Templates for Recognition of Objects with Regular Structure. In Joachim Weickert, Matthias Hein, and Bernt Schiele, editors, *Pattern Recognition*, Lecture Notes in Computer Science, pages 364–374, Berlin, Heidelberg, 2013. Springer.

[75] Lijun Wang, Jianming Zhang, Oliver Wang, Zhe Lin, and Huchuan Lu. SDC-Depth: Semantic Divide-and-Conquer Network for Monocular Depth Estimation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 538–547, Seattle, WA, USA, June 2020. IEEE.

[76] Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. Tartanair: A dataset to push the limits of visual slam. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4909–4916. IEEE, 2020.

[77] Daitao Xing, Jinglin Shen, Chiuman Ho, and Anthony Tzes. ROIFormer: Semantic-Aware Region of Interest Transformer for Efficient Self-Supervised Monocular Depth Estimation, December 2022. arXiv:2212.05729 [cs].

[78] Chenxiao Yang, Junwei Pan, Xiaofeng Gao, Tingyu Jiang, Dapeng Liu, and Guihai Chen. Cross-Task Knowledge Distillation in Multi-Task Recommendation. *AAAI*, 2 2022.

[79] Zhendong Yang, Ailing Zeng, Zhe Li, Tianke Zhang, Chun Yuan, and Yu Li. From knowledge distillation to self-knowledge distillation: A unified approach with normalized loss and customized soft labels. *ICCV*, 2023.

[80] Han Jia Ye, Su Lu, and De Chuan Zhan. Distilling cross-task knowledge via relationship matching. In *CVPR*, 2020.

[81] Mingkuan Yuan and Yuxin Peng. CKD: Cross-task knowledge distillation for text-to-image synthesis. *IEEE Transactions on Multimedia*, 8:1955–1968, 2020. Conference Name: IEEE Transactions on Multimedia.

[82] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *ICLR*, 2019.

[83] Amir Zamir, Alexander Sax, William Shen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling Task Transfer Learning. *arXiv:1804.08328 [cs]*, April 2018. arXiv: 1804.08328.

[84] Borui Zhao, Renjie Song, and Yiyu Qiu. Decoupled Knowledge Distillation. *CVPR*, 2022.

[85] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene Parsing Through ADE20K Dataset. In *CVPR*, page 9, 2017.

[86] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. *CVPR*, 3 2017.

[87] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A Comprehensive Survey on Transfer Learning. *arXiv preprint*, 2020.