Sharing Decoders: Network Fission for Multi-task Pixel Prediction Supplementary

1. Backbone Architecture

1.1. Server Model

We also use two auxiliary loss branches at the end of the first and second stages of the deconvolutional layers just as in [8]. Each branch decreases the features to the number of labels using 1x1 convolution with batch normalization and then bilinear upsampling to match the input resolution. We consider the tasks of semantic labeling, depth prediction, and surface normal prediction. For all equations below, α_1 is 0.6, and α_2 is 0.5 for the auxiliary loss balancing. The encoder is ResNet-50[3] with full pre-activation residual units[4] and multi-scale residual units [8] with varying dilation rates. After the encoder, we use the efficient atrous spatial pyramid pooling module (eASPP) [8] to create the bottleneck. The output of this is 16-times down-sampled.

For the decoder, we use three stages comprising of convolutional and deconvolutional layers. The first stage is upsampled by a factor of 2. The second stage concatenates those results with the first skip refinement from the encoder with a 1x1 convolution. That result is passed through two 3x3 convolutions followed by a deconvoluional layer that upsamples by a factor of 2. The second stage is the same as the second stage except using the 2nd skip refinement from the encoder. The output is then finally used as the input to a 1x1 convolutional transpose to reduce the number of feature channels to the desired output for the task which is then upsampled 4x to match the input resolution. This is better seen in code[1]. We find for multi-task learning, the auxiliary branch losses help improve results, discussed in Section 3.

1.2. Mobile models

The mobile models are very similar in structure to the server model. We use the same number of branches, u-net decoder layers, auxiliary losses, and loss balancing. The differences are the encoders being variants of Mobilenet and mobilenet v3 like discussed in the paper. Full mobile diagrams are included as separate PDFs with this supplementary.

Both mobile models are float16 quantized with TFLite. With the mobile models, we predict disparity (scale-shift invariant inverse depth) instead of absolute depth since absolute depth is a more difficult task and it doesn't scale well to many individual devices such as mobile phones with several different cameras. However, there is an issue with scale-shift invariant depth in that it is not guaranteed to be in any set range. Since float16 needs to be below 65536, this can cause some errors if the values are too high (or too low if they go below the precision limit). To solve this, we also in 0.25 * the depth loss described in the paper to keep the predictions in a valid range.

1.3. Architecture Hyper-parameters

We use the same training hyper-parameters to allow for fair evaluation. For the synthetic Scenenet ablation studies, we use a batch size of 16 320x240 images for 150,000 iterations with no data augmentation as Scenenet is considerably large and we wished to remove the effect data augmentation can have on training. For the real data, we use an input of 768x384 with a batch size of 8 for single tasks and 6 for multi-task with data augmentations of random flips, crops, and lighting changes used. We use a learning rate of 0.001 that has a polynomial decay with a decay step of 30,000 and a decay power of 0.9. We use a weight decay of 0.0005 and batch normalization decay of 0.99. We use an Adam solver with $\beta_1 = 0.9$ and $\beta_1 = 0.999$. The eASPP parameters are the same as in [8] with an eAspp rate of 3, 6, and 12 for the three stages. For the four encoder stages, we use residual units of 3, 4, 6, and 3 and a filter size of 256, 512, 1024, and 2048 with strides of 1, 2, 2, and 1 respectively. We initialize our encoder with ImageNet weights in our experiments but for the newly learned layers we use He initialization [3] (see code [1]).

2. Metrics

For depth metrics, we use the well established metric in Equation 1 where δ is 1.25, 1.25^2, and 1.25^3. y is the ground truth pixel-wise depth and z is the final output of the task decoder. Some papers also include relative depth and root-mean squared error which are also easy to compute but for ease of reading we have avoided including these.

$$max(\frac{y}{z}, \frac{z}{y}) < \delta \qquad (1)$$

Single-Task	Normals % <			Depth % <			Semantics
Method	11.25	22.5	30	1.25	1.25^2	1.25^3	mIoU
Normals	83	91.5	93.8	-	-	-	-
Depth	-	-	-	86.1	95.3	97.5	-
Semantics	-	-	-	-	-	-	50.3

Table 1: Here are the individual results for each modality on the Scenenet dataset. For this, each method is trained and evaluated on only one task which is why we indicate - on the others. On each of these results, higher is better.

For surface normal metrics, we use the metrics in Equation 2 where δ is 11.25, 22.5, and 30. $\hat{y}$ is the ground truth pixel-wise surface normal and $\hat{z}$ is the final output of the task decoder both normalized to a unit vector where each normal is also clipped between -1.0 and 1.0 in case of rounding errors. We also consider mean average error which is simply the mean of the result of the left hand side of the equation (ignore δ) across all valid pixels.

$$\frac{180}{\pi} \arccos \left(\sum_{i=0}^{3} \hat{y}^{(i)} \hat{z}^{(i)}\right) < \delta \tag{2}$$

For semantic labeling metrics, we use the mean intersection over union shown in Equation 3. y is the ground truth pixel-wise depth and z is the final output of the task decoder. This is common and we will not discuss it at length here.

$$\text{mIoU} = \sum_{i=0}^{n} \frac{y_i \cap z_i}{y_i \cup z_i} \tag{3}$$

2.1. Datasets

Our ablation studies are done on the **Scenenet RGB-D** dataset [6]. Scenenet RGB-D is a set of 5 million synthetic 320x240 RGB-D images from more than 15,0000 trajectories of synthetic layouts with random object poses with random lighting and texture synthesis. It's excellent for testing our method as the tasks predicted (semantic labels, depth, and surface normals) are vulnerable to frequent noise in real-world datasets where the depth data is collected from noisy time-of-flight sensors, the normals are calculated from the depth, and the semantic labels are often crowd-sourced, with propagated errors. To create the fairest ablation environment, we use no data augmentation with the all of the same hyper parameters and the given image size of 320x240 as input to our network.

We also use two real-world datasets to verify that our method works well on real-world data. For the surface normals of the real-world data, it is important the ground truth is accurate, and we use the method [5] described in Section 2.2 to calculate these.

Scannetv2 [2] is a dataset of approximately 2 million 1296x968 RGB images with 640x480 depth sensor images with included semantic segmentation labels. These labels were annotated in 3D and then back-projected into 2D, which allows for more labeled images but the annotations can be less accurate. For the semantic labels, we use the top 20 of the NYU40 split. We resize the RGB and depth images to 768x384 as in [8] as to fit onto a single Titan X GPU with a batch size of 8. For multi-task training, we reduce the batch size to 6 as the early-fission architecture doesn't fit with 8. We then train using 8 titan GPUs using synchronized training.

NYUDv2 [7] is a dataset of indoor environments taken with a Kinect device with 640x480 RGB-D pairs. Semantic labels are created for 1449 of these images, split into predefined train and test sets. There are 13 and 40 class label splits that literature uses for comparisons.

2.2. Surface Normal Ground Truth

To create ground truth surface normals, we use the method from [5] as it generates clean, semantically corrected surface normals. We use the parameters recommend by the authors, which are a depth in-painting window-size of 5, a normal max depth change factor of 0.02, a normal adaptive smoothing window of 10 for synthetic data and 30 for real-world data, and a planar threshold parameter of 0.4, which controls whether semantic planar surfaces are joined.

3. Auxiliary Loss Ablation Study

Recall the loss equations from the paper where each loss for the tasks has two auxiliary losses weighted with α_1 and α_2. Given that these are computed at the two stages of the decoder that are shared between tasks in mid fission, it is not clear whether these would help or cause task interference. In this study shown in Table 2, we aim to explore this issue. We evaluate the mid fission strategy without loss balancing using the same mid fission architecture and evaluation described in the Fission-scheme Ablation Study section of the paper. Results considering both loss balancing and depth are shown in Tables 54.

To evaluate this, we test four different auxiliary loss strategies: no auxiliary losses denoted $\alpha = 0$ ($\alpha_1 = 0$,$\alpha_2 = 0$ for both normals and semantic labels), both auxiliary losses denoted $\alpha_* = 0.6, 0.5$ ($\alpha_1 = 0.6$,$\alpha_2 = 0.5$ for both semantics and normals), only auxiliary loss for normals denoted as $\alpha_s = 0$ ($\alpha_1 = 0.6$,$\alpha_2 = 0.5$ for normals,$\alpha_1 = 0$,$\alpha_2 = 0$ for semantic labels), and only auxiliary loss for semantic labels denoted as $\alpha_n = 0$ ($\alpha_1 = 0.6$,$\alpha_2 = 0.5$ for semantic labels,$\alpha_1 = 0$,$\alpha_2 = 0$ for normals). We do this for both semantic label initialization and surface normal initialization as this can have a joint effect on training.

Method	Normals				Semantics
	$\% < 11.25$	$\% < 22.5$	$\% < 30$	MAE	mIoU
Individually Trained Baselines					
Normals	83	91.5	93.8	8.3	-
Semantics	-	-	-	-	50.3
Initialized from Labels					
All $\alpha = 0$	70.1	87.2	91.4	11.9	51.7
$\alpha_* = 0.6, 0.5$	71.3	87.5	91.6	11.6	51.6
$\alpha_s = 0$	76	89.1	92.5	10.4	52.2
$\alpha_n = 0$	61.9	84.4	89.8	13.6	51.7
Initialized from Normals					
All $\alpha = 0$	77.4	89.8	92.9	10.1	50.8
$\alpha_* = 0.6, 0.5$	79.3	90.5	93.5	9.4	51.8
$\alpha_s = 0$	81.9	91	94	8.7	50.4
$\alpha_n = 0$	75.8	89.5	92.9	10.4	51.7

Table 2: Joint Normals and Semantics results with different aux losses using our proposed mid fission without loss balancing.

Method	Normals				Semantics
	$\% < 11.25$	$\% < 22.5$	$\% < 30$	MAE	mIoU
Individually Trained Baselines					
Normals	83	91.5	93.8	8.3	-
Semantics	-	-	-	-	50.3
Initialized from Labels					
$\lambda_n = 1.0$	71.3	87.5	91.6	11.6	**51.6**
$\lambda_n = 5.0$	74.5	88.6	92.2	10.8	49.1
$\lambda_n = 10.0$	78.2	89.8	92.9	9.8	49
$\lambda_n = 15.0$	78.9	90.1	93.2	9.6	48.7
Initialized from Normals					
$\lambda_n = 1.0$	79.3	90.5	93.5	9.4	**51.8**
$\lambda_n = 5.0$	80.5	91	93.8	9.2	**51.8**
$\lambda_n = 10.0$	**86.6**	**93.4**	**95.3**	**7**	**51.2**
$\lambda_n = 15.0$	**86.6**	**93.4**	**95.3**	**7**	50

Table 3: Joint Normals and Semantics results with different loss balancing using our proposed mid fission.

Surprisingly, the semantic mean IoU is mostly unchanged when initialized from the single task trained solely on semantic labels, where the highest accuracy is actually where only surface normal auxiliary losses are used. The semantic prediction is even higher than our results in Table 1 in the paper but the surface normal prediction is significantly worse. When initialized from the single task trained solely on surface normals, surface normal metrics are much closer to the single task results whereas semantic label predictions are still outperforming the single task results and come close to results when initialized by label prediction. Our hypothesis for this is that surface normals are a better initialization task given their reliance on edge and surface based features.

4. Loss Balancing Ablation Study

Empirical tests show that the semantic cross entropy loss is approximately 10x the surface normal cosine loss so we test loss balancing in Table 3. Again we evaluate the mid fission strategy using the same architecture and evaluation described in thie Fission Study in the main paper, with the auxiliary loss for both surface normals and semantic labels as described in Section 3. We evaluate with $\lambda_{\text{semantics}} = 1$ in all cases and $\lambda_{\text{normals}} = \{1, 5, 10, 15\}$ denoted as λ_n. Balancing with other modailties is shown in Table 4 and Table 5.

Unsurprisingly, when initializing with labels, increasing the λ_n of the cosine loss for surface normals results in better surface normals but makes the semantic prediction underperform single-task prediction. However, when initializing with normals, semantic accuracy continues to outperform single task prediction (though decreasing slightly as λ_n increases), whereas surface normal prediction starts to outperform single task prediction. Note that at $\lambda_n = 15$, surface normal prediction remains the same but semantic labeling metrics decrease. Therefore, the choice of $\lambda_n = 10$ is validated here with greater than 4% more pixels being under 11.25 degrees error and almost 2% more mIoU compared to the single-task predictions.

Method	Normals % <			Depth % <		
	11.25	22.5	30	1.25	1.25^2	1.25^3
Baseline						
Normals	83	91.5	93.8	-	-	-
Depth	-	-	-	**86.1**	**95.3**	**97.5**
No Loss Balance						
Early	79.2	91.2	93	82.9	91.9	94.2
E-Mid	78.8	89.9	92.9	82.0	89.5	91.8
Mid	80.1	92	93.3	80.9	89.9	92.4
Late	76.6	89.2	92.4	81.6	89.1	91.4
$\lambda_n = 10$, $\lambda_d = 1$						
Early	83.6	92.2	94.5	81.1	91.5	94.3
E-Mid	85.5	92.9	94.9	83.5	92.5	94.7
Mid	**85.5**	**93.1**	**95.1**	81.8	90.9	93.2
Late	85	92.6	94.8	82.5	90.1	92.2

Table 4: Joint Normals and Depth results with different fission methods both with no loss balancing and when the loss is balanced ($\lambda_{normals} = 10$ and $\lambda_{depth} = 1$). For this ablation, auxiliary loses are used ($\alpha_1 = 0.6, \alpha_2 = 0.5$ for both tasks).

In Table 4, we show results for early, early-mid, mid, and late fission for a model trained jointly on surface normals and depth showing the impact of loss balancing. For early-mid and mid fission, we fine-tune from normals, otherwise we train from scratch as that doesn't help for early/late fission. Surface normal prediction for early, late, and mid fission improve when using loss balancing, which makes sense given the depth loss is approximately 20x the normals loss. Interestingly enough, depth gets slightly better for both late and mid fission even though the depth loss has a lower overall impact on the total loss. This verifies the importance of loss balancing for mid fission regardless of the tasks learned. Note that early-mid and mid fission generally perform the best here.

Initialization	Depth % <			Semantics
	1.25	1.25^2	1.25^3	mIoU
Baseline				
Depth	**86.1**	**95.3**	**97.5**	-
Semantics	-	-	-	50.3
No Loss Balance				
Early	39.2	64.8	79.1	48.5
E-Mid	73.7	90.7	95.2	50.0
Mid	80.1	93.1	96.5	49.2
Late	84.7	95.2	97.5	48.6
$\lambda_s = 2$, $\lambda_d = 1$				
Early	66	86.8	93.3	50.2
E-Mid	81.7	94.1	**97.1**	**51.2**
Mid	82.8	**94.7**	**97.3**	**51.0**
Late	82.3	**94.6**	**97.2**	48.7

Table 5: Joint Depth and semantic label results with different fission methods with no loss balancing and when the loss is balanced ($\lambda_{semantics} = 2$ and $\lambda_{depth} = 1$). For this ablation, auxiliary loses are used ($\alpha_1 = 0.6, \alpha_2 = 0.5$ for both tasks).

In Table 5, we show results for early, early-mid, mid, and late fission for a model trained jointly on semantic labels and depth showing the impact of loss balancing. For early-mid and mid fission, we fine-tune from normals, otherwise we again train from scratch as we find that didn't help for early/late fission just as shown in the main paper. Semantic label prediction for all fission methods improve when using loss balancing, which makes sense given the depth loss is approximately 2x the semantic labels cross-entropy loss. Depth gets slightly better for both early, early-mid, and mid fission. This again shows the importance of loss balancing for mid fission regardless of the tasks learned. Mid fission is always improved by balancing the losses. Early fission has very bad results for depth with these joint tasks. Our hypothesis is that there are not many good task-shared features between depth and semantic labels in the bottleneck generated by the encoder. Note that early-mid and mid fission generally perform the best here.

Balancing with all 3 modalities is shown in the main paper but here we consider only surface normals and semantic labels as in the previous sections. Unsurprisingly, when initializing with labels, increasing the λ_n of the cosine loss for surface normals results in better surface normals but makes the semantic prediction under-perform single-task prediction. However, when initializing with normals, semantic accuracy continues to outperform single task prediction (though decreasing slightly as λ_n increases), whereas surface normal prediction starts to outperform single task prediction. Note that at $\lambda_n = 15$, surface normal prediction remains the same but semantic labeling metrics decrease. Therefore, the choice of $\lambda_n = 10$ is validated here with greater than 4% more pixels being under 11.25 degrees error and almost 2% more mIoU compared to the single-task predictions.

5. Initialization Study

Initialization	Normals % <			Depth % <		
	11.25	22.5	30	1.25	1.25^2	1.25^3
Scratch	66.4	86.2	90.7	80.8	89.4	92
Normals	**85.5**	**93.1**	**95.1**	81.8	90.9	94.1
Depth	74.5	88.4	91.9	**83.9**	**91.9**	**94.1**

Table 6: Joint Normals and Depth results using mid fission with loss balancing ($\lambda_{normals} = 10$ and $\lambda_{depth} = 1$). Here we compare when trained from 3 different initializations: Scratch being ImageNet fine-tuning, Normals being initialized with the network trained on normals, Depth being initialized with the network trained on depth. For this ablation, auxiliary loses are used ($\alpha_1 = 0.6, \alpha_2 = 0.5$ for both tasks).

In Table 6, we show results for mid fission for a model trained jointly on surface normals and depth with different initializations. We initialize in three different ways: from scratch, from the normals network, and from the depth network. Note that for mid fission, as expected, initializing from normals improves normals substantially and improves depth some, while initializing from depth improves depth some and improves normals some. Given this, initializing from normals still seems to be the better method. However, it is not as clear cut as the joint tasks of surface normals and semantic labels. In that case, both tasks are the highest when initialized with normals. Our hypothesis is that surface normals are a better initialization method for the tasks in our multi-task model.

Initialization	Depth % <			Semantics
	1.25	1.25^2	1.25^3	mIoU
Scratch	33.5	60.8	76.7	**50.4**
Labels	40.8	65.2	80.1	**50.9**
Depth	77.2	91.4	95.5	43.1
Normals	**82.8**	**94.7**	**97.3**	**50.4**

Table 7: Joint Semantics and Depth results using mid fission with loss balancing ($\lambda_{semantics} = 2$ and $\lambda_{depth} = 1$). Here we compare when trained from 3 different initializations: Scratch being ImageNet fine-tuning, Normals being initialized with the network trained on normals, Depth being initialized with the network trained on depth. For this ablation, auxiliary loses are used ($\alpha_1 = 0.6, \alpha_2 = 0.5$ for both tasks).

In Table 7, we show results for mid fission for a model trained jointly on semantic labels and depth with different initializations. We initialize in three different ways just as in Table 6. Note that for mid fission, as expected, initializing from labels improves labels slightly and improves depth some, while initializing from depth improves depth substantially but semantics drastically decreases accuracy. This again gives credence to our hypothesis that there are not as many task-shared features between depth and semantic labels. Interestingly enough, initializing from normals improves metrics for both even though it isn't a predicted task here. This confirms that initializing the network by training a method on surface normals creates good features for several related tasks. Given normals were seen as a very good task for transfer learning in Taskonomy [9], this makes some amount of sense.

5.1. Variability Ablation Study

Method	Normals				Semantics
	% < 11.25	% < 11.25	% < 30	MAE	mIoU
Run 1	86.6	93.4	95.3	7	51.2
Run 2	86.6	93.5	95.3	7	51.4
Run 3	86.6	93.4	95.3	7	51.3
Range (+/-)	0.05	0.07	0.05	0.02	0.1

Table 8: 3 Different training runs of our mid-fission network on normals and semantics with normal initialization and balanced losses.

As shown in Table 8, we find our metrics only shift +/-0.1 at most verifying these results are meaningful and not just noise. We did this on the ablation study on Scenenet using the server model with the same experimental methodology used to create Table 1 in the main paper.

References

[1] Paper code. `https://github.com/StevenHickson/Adapnet-Shape`.

[2] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas A Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, volume 2, page 10, 2017.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[5] Steven Hickson, Karthik Raveendran, Alireza Fathi, Kevin Murphy, and Irfan Essa. Floors are flat: Leveraging semantics for real-time surface normal prediction. In *The IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2019.

[6] 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 *Proceedings of the International Conference on Computer Vision (ICCV)*, volume 4, 2017.

[7] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[8] Abhinav Valada, Rohit Mohan, and Wolfram Burgard. Self-supervised model adaptation for multimodal semantic segmentation. *arXiv preprint arXiv:1808.03833*, 2018.

[9] Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3712–3722, 2018.