A. Training hyperparameters

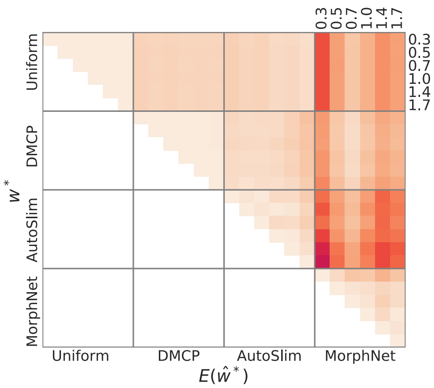
We used PyTorch [21] as our deep learning framework. We largely follow [22] for training hyperparameters. Specifically, learning rate grows linearly from 0 to $0.2s$ within the first 5 epochs from 0 where s depend on batch size B, *i.e.*, $s = \frac{B}{256}$. We use batch size of 1024 and distributed training over 8 GPUs and we have not used synchronized batch normalization layers. We set the training epochs to be 100. For optimizers, we use stochastic gradient descent (SGD) with 0.9 Nesterov momentum. As for data augmentation, we have adopted 0.1 label smoothing, random resize crop, random horizontal flops, and RandAugment [4] with parameter $N = 2$ and $M = 9$ following common practice in popular repository[3]. Note that these training hyperparameters are fixed for all experiments.

For hyperparameters specific to width optimization algorithms, we largely follow the hyperparameters used in respective methods. Specifically, we use 40 epochs to search for optimized width for all three algorithms. We enlarge the network by $1.5\times$ for DMCP and AutoSlim. Since MorphNet has FLOPs-aware regualrization, we normalize the FLOPs for each network and use $\lambda = 1$ for all experiments.

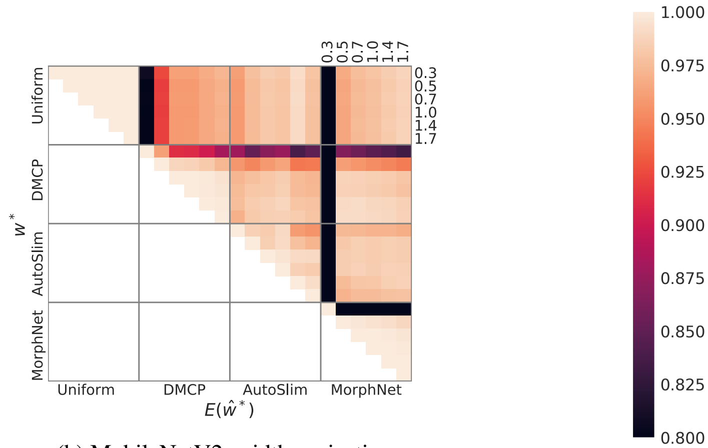
B. Similarity among width multipliers

In Section 4, we have analyzed the similarity between $\boldsymbol{w}^*$ and $\hat{\boldsymbol{w}}^*$ in the accuracy space. Here, we show that $\boldsymbol{w}^*$ and $\hat{\boldsymbol{w}}^*$ are in fact similar in the vector space using cosine similarity.

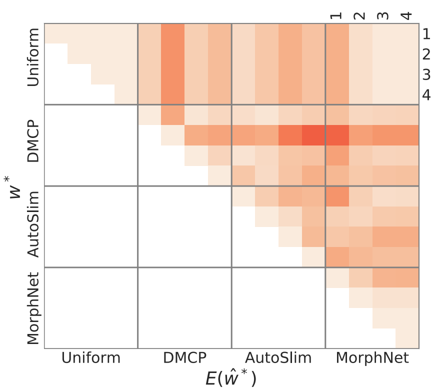
[3] https://github.com/rwightman/pytorch-image-models (We use their implementation for RandAugment and use 'rand-m9-mstd0.5' as the value for the 'aa' flag.



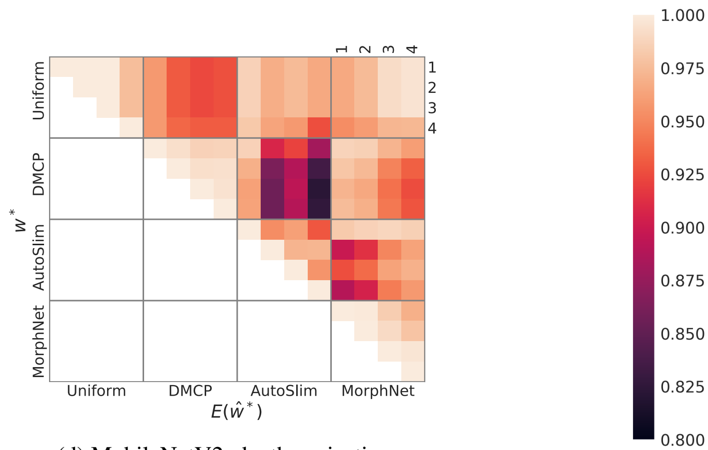
(a) ResNet18, width projection



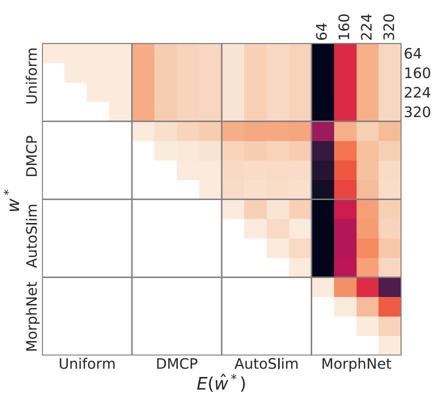
(b) MobileNetV2, width projection



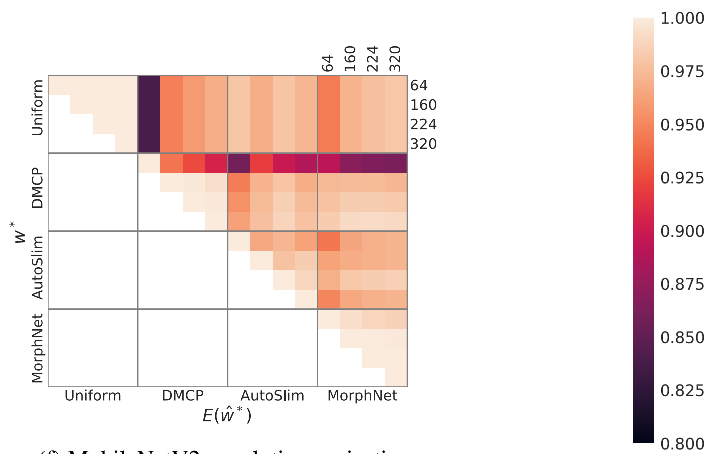
(c) ResNet18, depth projection



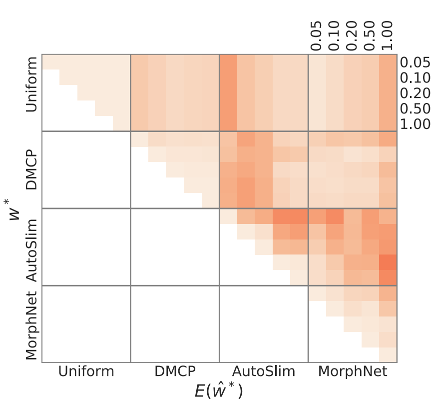
(d) MobileNetV2, depth projection



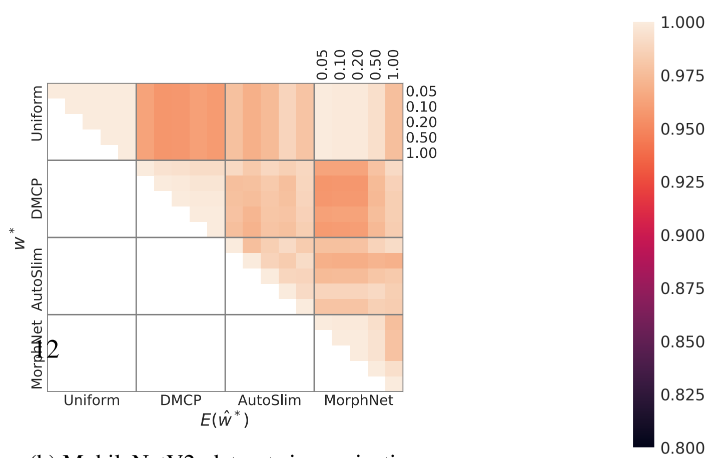
(e) ResNet18, resolution projection



(f) MobileNetV2, resolution projection



(g) ResNet18, dataset size projection



(h) MobileNetV2, dataset size projection