ETHZ CIL 2025: Uncertainty-Aware Ensemble for Monocular Depth Estimation

Martin Wertich [*1] **Dmitry Knorre** [*1] **Eleftheria Vrachoriti** [*1] **Sergejs Zahovskis** [*1]

Abstract

In this work, we explored whether the performance of the state-of-the-art models on the Monocular Depth Estimation task can be improved by averaging multiple models' predictions based on the model's uncertainty. To verify this assumption, we fine-tuned a Dense Prediction Transformer model modified to predict uncertainty using Gaussian Negative Log Likelihood loss. Then, we trained multiple experts by further fine-tuning the aforementioned model for different room types. To combine their predictions, we trained a separate meta-model that takes the models' uncertainty into account. We showed that this approach decreases siRMSE compared to the performance of the base model. Additionally, we identified that this approach is sensitive to hyperparameters and can suffer from mode collapse. Finally, our results show that the meta-model regularly combines the ensemble's individual predictions in a manner consistent with the intuition of the expert's room-type domain knowledge and generalizes to ambiguous images.

1. Introduction

Monocular Depth Estimation (MDE) is a task of estimating the distance from the camera to each pixel from a single captured RGB image. Today, deep learning-based approaches are widely used for MDE and are known to be very efficient (Ming et al., 2021). However, their biggest downside lies in their very low interpretability. A common approach to overcome this problem is to measure the uncertainty of the model. The uncertainty was shown to correlate with the quality of the model predictions (ŞAHiN et al., 2024). In this work, we explored the way to leverage this fact in order to improve the model's performance. To this end, we combine the model's uncertainty estimates with ensembling, as a common technique to boost the model's performance (Ganaie et al., 2022). To test it, we developed a pipeline based on stacking several models for MDE based on their uncertainty. To be specific, we completed the following steps:

1. We fine-tuned the pre-trained Dense Prediction Transformer (DPT) (Ranftl et al., 2021) on available data to predict uncertainty using the Gaussian Negative Log Likelihood Loss Function (Nix & Weigend, 1994). We will refer to it as our **base model**.
2. We trained several experts by fine-tuning them further on a subset of our training data. All experts were trained to predict their uncertainty, just as the base model.
3. We developed a **meta-model** that predicts the weights to sum up the experts and the base model predictions based on their uncertainty.

Throughout the experiments, we observed that our approach leads to a noticeable improvement compared to both the base model and individual expert models. The code for the project is available at: `https://github.com/mwertich/CIL`.

2. Related Work

Since the introduction of the first neural network-based approach for MDE in (Eigen et al., 2014) the deep learning paradigm has dominated the field ever since. Thus, recent CVPR challenges were dominated by works that fine-tuned the state-of-the-art models for MDE on the provided dataset (Obukhov et al., 2025). Among such models, one that can be easily fine-tuned for a new MDE dataset is Dense Prediction Transformer (DPT) (Ranftl et al., 2021), which utilizes the Vision Transformer architecture (Dosovitskiy et al., 2020).

The estimation of a model's uncertainty in its predictions has been an area of active research. In (Landgraf et al., 2025), it was shown that the standard method in the MDE setting is training a model to predict the variance of its prediction using the uncertainty-aware Gaussian Negative Log-Likelihood (GNLL) loss function, which was proposed in (Nix & Weigend, 1994).

[*]Equal contribution [1]Department of Computer Science D-INFK, ETH Zurich, Zurich, Switzerland. Correspondence to: Martin Wertich <mwertich@student.ethz.ch>, Dmitry Knorre <dknorre@student.ethz.ch>, Eleftheria Vrachoriti <evrachoriti@student.ethz.ch>, Sergejs Zahovskis <sergejsz@student.ethz.ch>.

Computational Intelligence Lab FS2025, ETH Zuerich

Ensemble learning is a common technique aimed at improving model performance in many settings, including MDE (Ali et al., 2023). For example, in (Mitra et al., 2024), the authors showed that making predictions using a weighted sum of the MiDaS and MonoDepth2 predictions can outperform both models used separately. Further development of ensemble learning is the use of a separate neural network to process the ensemble components' predictions. Thus, in (Shao et al., 2023), the authors train multiple neural networks with different architectures, which are further processed by a recurrent neural network to predict a final depth estimation.

An idea to utilize models' uncertainty to learn an ensemble on top of them is also present in the literature. In (Bae et al., 2022), the authors use multiple networks to predict surface normals as well as depth and uncertainties of these predictions. They further put all this through a gated recurrent unit to iteratively improve the depth prediction. Finally, in (Zhang et al., 2022), the authors fuse the depth prediction of the echo-based and vision-based models with the weight inversely proportional to the predicted variance of the model prediction.

In this work, we aim to investigate whether models that share the same architecture but were trained on different data could be used to create an uncertainty-driven ensemble for MDE.

3. Methodology

We built our work on top of the Dense Prediction Transformer (DPT) model from the MiDaS model family (Ranftl et al., 2021), which shows high performance and generalizes well for new datasets.

3.1. Uncertainty Quantification

We extend the DPT model architecture with an uncertainty quantification mechanism to generate more robust and interpretable depth predictions. To achieve this, we build on top of the pre-trained DPT backbone and follow a dual-head approach, as presented in (Landgraf et al., 2024).

Alongside the standard depth prediction head, we introduce a second head that estimates per-pixel uncertainty for the corresponding depth prediction in the form of variance. We refer to the resulting model as the base model.

The depth prediction head uses a ReLU activation function to ensure non-negative predicted depth values, while the uncertainty head utilizes a Softplus activation function to force strictly positive variance estimates. Similar to the approach described in (Landgraf et al., 2025), we train base model using the GNLL loss (Nix & Weigend, 1994) for the joint estimation of depth and uncertainty, defined as follows:

$$\mathcal{L}_{\text{GNLL}} = \frac{1}{2}\left(\frac{(d-\hat{d})^2}{\sigma^2} + \log \sigma^2\right) \tag{1}$$

where $\hat{d}$ and σ^2 are the predicted depth and variance, respectively, and d is the ground truth depth.

3.2. Expert Models

Since our data consists of only indoor scenes, we categorized images based on room types using a pre-trained ResNet18 classifier, which received 56.51 Top-1-Accuracy and 86.00 Top-5-Accuracy in the Places365 standard data set (López-Cifuentes et al., 2020).

Individual room-type-specific experts are obtained by further fine-tuning the base model only on the images of the respective room-type category. This was motivated by the fact that images of the same category share more characteristics and features than images from other categories.

3.3. Meta-model Ensembling

We define a discrete index set $I = [0, 1, ..., K]$, in which index 0 represents the base model ($k = 0$) and the other indices the respective expert $k \in [1, 2, ..., K]$.

We introduced a separate meta-model to automatically combine the predictions of the base and expert models. For the meta-model, we used a standard U-Net architecture (Ronneberger et al., 2015), which takes the image and the uncertainty of the base model as input and outputs a logit tensor z (see Table 3). The output logit z_{ij}^k for model $k \in I$ and a input image pixel x_{ij} expresses how strongly the base model ($k = 0$) or the respective expert $k \in [1, 2, ..., K]$ should be weighted for the final prediction.

Let $\hat{d_{ij}}^k$ be the predicted depth and $(\sigma_{ij}^2)^k$ the uncertainty (variance in GNLL loss) at pixel x_{ij} for model $k \in I$. To obtain a linear combination for each pixel, we apply SoftMax with a temperature τ and divide by the respective model's prediction variance (see Equation (2)).

$$\lambda_{ij}^k = \frac{\exp\left(\frac{z_{ij}^k}{(\sigma_{ij}^2)^k \cdot \tau}\right)}{\sum_{l \in I} \exp\left(\frac{z_{ij}^l}{(\sigma_{ij}^2)^l \cdot \tau}\right)} \quad \rightarrow \quad \hat{d_{ij}} = \sum_{k \in I} \lambda_{ij}^k \cdot d_{ij}^k$$

$$\lambda_{ij}^k \geq 0, \quad \sum_{k \in I} \lambda_{ij}^k = 1, \quad \forall i, j \tag{2}$$

The inverse variance scaling incorporates the uncertainty of the base model and the expert into the framework. The idea behind this is that the quality of the model predictions is higher if the uncertainty of the predictions is lower.

The meta-model is trained with the following loss function

consisting of three components in Equation (3).

$$l(\hat{d}, \lambda, d) = \sum_{i,j} \Big[\alpha(\hat{d}_{ij} - d_{ij})^2 + \beta\, \mathrm{H}(\lambda_{ij}^k, O_{ij}^k) + \gamma\, \mathrm{H}(\lambda_{ij}^k)\Big]$$
$$O_{ij}^k = \arg\min_k \left(|d_{ij}^k - d_{ij}|\right) \qquad \alpha, \beta, \gamma \geq 0 \quad (3)$$

H(X) and H(X, Y) denote the entropy and the cross-entropy of discrete variables X and Y. The first term is the mean squared error between the meta-model prediction $\hat{d}_{ij}$ and the ground truth d_{ij}. The second term is the cross-entropy between the estimated percentages λ_{ij}^k in the linear combination and the best optimal model O_{ij}^k for that pixel. The intuition behind this choice is to encourage the meta-model to choose the strongest expert for that pixel. The last term acts as a regularizer and is the entropy of λ_{ij}^k, to discourage predicting overly flat linear combinations and to encourage the exploration of different experts' predictions.

4. Experiments

We divided the total of 23971 images from the dataset into 80% training (19176) and 20% validation images (4795). We trained each of the three consecutive steps with the Adam optimizer (Kingma & Ba, 2017).

Base Model Training.We fine-tuned the pre-trained DPT model on our training set for 5 epochs with a learning rate of 10^{-5} and a batch size of 1. The loss function plateaued quickly, as shown in Figure 3.

Expert Training. Then we fine-tuned the five room-type experts with uncertainty for an additional 10 epochs using Adam with the same training parameters as the base model but a slightly lower learning rate of $2 \cdot 10^{-6}$.

Meta-model Training. Then we collected all the predictions and the uncertainty estimates for both the base model and all experts and trained the meta-model for 10 epochs with a learning rate of $2 \cdot 10^{-6}$. For our default meta-model experiment, we used a temperature of $\tau = 1$ to sharpen the predictions of linear combinations, and the parameters of the loss function in Equation (3) are $\alpha = 1$, $\beta = 0.01$, $\gamma = 10^{-4}$. Due to limited space on the cluster, we split our 19176 training images into two halves. We trained the meta-model for five epochs on each of the respective halves, for which the results are summarized over the entire trajectory.

All reported scores were obtained as the average over three runs with different seeds.

5. Results and Discussion

The models are evaluated based on the Scale-Invariant RMSE (siRMSE) score, as specified in ETHZ CIL Monocular Depth Estimation 2025 (Kaggle).

5.1. Base Model training

Before fine-tuning, the pre-trained DPT model was getting a siRMSE of 6.1925. After fine-tuning, the base model significantly reduced the error to siRMSE of 0.0670.

To evaluate the quality of predicted uncertainty estimates, we calculated three metrics proposed in (Mukhoti & Gal, 2018): **PA** $= 0.994$, **PU** $= 0.869$, and **PAvPU** $= 0.5165$. This corroborates our assumption that the uncertainty correlates with the prediction error. For more information on how to calculate these metrics, see Appendix C. A visualization of the uncertainty and depth error maps, as shown in Figure 4-Figure 5, corroborates the results qualitatively.

5.2. Expert Models

After fine-tuning the five room-type experts, we received the following siRMSE scores by evaluating them both on their respective categories and the entire validation set (see Table 1). We can observe an overall improvement of an siRMSE between 0.01 and 0.015 on their respective category. The siRMSE on the full dataset changes by less than 0.005 compared to the siRMSE of the base model of 0.0670. The remaining evaluation metrics are illustrated in Table 4.

Category	Epoch 0, category val. set	Epoch 10, category val. set	Epoch 10, complete val. set
Sleeping	0.0484	0.0371	0.0720
Living	0.0666	0.0537	0.0692
Kitchen	0.0735	0.0604	0.0721
Work	0.0629	0.0489	0.0692
Remaining	0.0876	0.0721	0.0651

Table 1. siRMSE scores before and after fine-tuning each expert on the respective category, evaluated on the respective category and the entire validation set

5.3. Meta-model

After training the meta-model on the entire dataset, the final validation siRMSE is the first entry in Table 2. We also illustrate training loss and validation siRMSE scores in Figure 10.

5.3.1. Comparative Analysis

We compare the meta-model ensemble against the **three** baselines: *Base model*, *Trivial Ensembling* and *Inverse Variance-Weighted Averaging*, as well as **two** oracle-based approaches: *Category Oracle* and *Optimal Oracle*.

Base model. The first is the sole prediction of the base

model itself.

Trivial Ensembling. In this approach, we take the prediction of the base model and all experts and average them with equal weights.

Inverse Variance-Weighted Averaging. In this approach, we sum up the predictions of all models with the weights inversely proportional to the uncertainty estimates predicted by these models (Equation (4)). This approach is inspired by the work (Zhang et al., 2022).

$$\hat{d_{ij}} = \sum_{k \in I} \left(\frac{\exp\left(\frac{1}{(\sigma_{ij}^2)^k}\right)}{\sum_{l \in I} \exp\left(\frac{1}{(\sigma_{ij}^2)^l}\right)} \cdot d_{ij}^k \right) \tag{4}$$

Category Oracle & Optimal Oracle. These two oracles analyze the theoretical capabilities of the meta-model. For the category oracle, we consider that the model picks the expert of the category to which the image has been assigned according to Appendix A, so setting $\lambda_{ij}^l = 1$, where l is the expert to which the input image x has been assigned. For the optimal oracle, we compare against an overly optimistic oracle, which picks the best model for each pixel.

5.3.2. Discussion of Scores

All siRMSE scores are reported in Table 2 , and the full scores in Table 4. Our meta-model performs better than all three baselines.
The stronger oracle performance indicates that better hyperparameters or a meta-model architecture can theoretically improve the results.

Method	siRMSE scores
Our meta-model	**0.0601**
Base model	0.0670
Trivial Ensembling	0.0633
Inverse Variance-Weighted Averaging	0.0627
Category Oracle	0.0549
Optimal Oracle	0.0405

Table 2. Evaluation meta-model on the entire validation set against three baselines and two oracle approaches

5.3.3. Mode Collapse

The training of the meta-model posed a significant challenge as it is heavily influenced by **Mode Collapse**.

Mode Collapse. This is by far the severest obstacle we have faced. The problem is that the model needs to select a convex linear combination weights λ_{ij} for each pixel x_{ij}. This, in turn, leads to two separate problems:

Sparse Uniform Predictions. The first issue is that the meta-model always picks the same model l (usually the base model $l = 0$), which means that λ_{ij}^l is large, as this model performs on average the strongest, even though another model would be better for specific pixel regions. This problem is more evident if the experts are too specialized and the model constantly falls back to the base model.

Mode Collapse occurs when the meta-model predominantly predicts the same expert for each pixel in an image or predicts the same expert for most images.

5.3.4. Prediction Maps Interpretability

We often observe that the chosen best expert is consistent with the intuition about the image scene layout (e.g., "kitchen" expert for all pixels in a table). We discuss it in more detail with the help of combination maps in Appendix D.

6. Conclusion and Future Work

In this work, we explored whether the uncertainty estimates can be used to enhance the performance of the pre-trained MDE models. To this end, we developed an uncertainty-aware ensemble technique. We used DPT model from the MiDaS family as our base model.

To explore the performance of our approach, we fine-tuned the DPT model with the capability to predict uncertainty estimates. This base model showed competitive metrics for the quality of uncertainty estimates. Then we further fine-tuned the base model on different subsets of the dataset to acquire the expert models, decreasing the siRMSE on these subsets by more than 0.01. Finally, we trained a U-net-based meta-model to combine the base and expert models' predictions.

The resulting pipeline showed an improvement in the siRMSE score compared to the base model and the two baseline ensembles: *Trivial Averaging* and *Inverse Variance-Weighted Averaging*.

Additionally, we identified that our approach is prone to mode collapse if the expert models are not distinct enough and therefore require careful tuning of the hyperparameters. Our results show that the combination maps can be interpreted as consistent: For images with distinctive features, the most strongly selected experts correspond to objects that appear frequently in the respective expert category, generalizing to ambiguous images.

We speculate that the proposed approach can generalize to other architectures and various types of experts. Moreover, we believe that using different state-of-the-art models as experts for MDE can further improve the performance of the proposed approach.

References

Ali, A., Ali, R., and Baig, M. Improving the quality of monocular depth estimation using ensemble learning. In *2023 6th International Conference on Information Systems and Computer Networks (ISCON)*, pp. 1–5. IEEE, March 2023. doi: 10.1109/iscon57294.2023.10112150. URL http://dx.doi.org/10.1109/ISCON57294.2023.10112150.

Bae, G., Budvytis, I., and Cipolla, R. Irondepth: Iterative refinement of single-view depth using surface normal and its uncertainty, 2022. URL https://arxiv.org/abs/2210.03676.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale, 2020. URL https://arxiv.org/abs/2010.11929.

Eigen, D., Puhrsch, C., and Fergus, R. Depth map prediction from a single image using a multi-scale deep network, 2014. URL https://arxiv.org/abs/1406.2283.

Ganaie, M., Hu, M., Malik, A., Tanveer, M., and Suganthan, P. Ensemble deep learning: A review. *Engineering Applications of Artificial Intelligence*, 115:105151, October 2022. ISSN 0952-1976. doi: 10.1016/j.engappai.2022.105151. URL http://dx.doi.org/10.1016/j.engappai.2022.105151.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization, 2017. URL https://arxiv.org/abs/1412.6980.

Landgraf, S., Hillemann, M., Kapler, T., and Ulrich, M. Efficient multi-task uncertainties for joint semantic segmentation and monocular depth estimation, 2024. URL https://arxiv.org/abs/2402.10580.

Landgraf, S., Qin, R., and Ulrich, M. A critical synthesis of uncertainty quantification and foundation models in monocular depth estimation, 2025. URL https://arxiv.org/abs/2501.08188.

López-Cifuentes, A., Escudero-Viñolo, M., Bescós, J., and García-Martín, Semantic-aware scene recognition. *Pattern Recognition*, 102:107256, June 2020. ISSN 0031-3203. doi: 10.1016/j.patcog.2020.107256. URL http://dx.doi.org/10.1016/j.patcog.2020.107256.

Ming, Y., Meng, X., Fan, C., and Yu, H. Deep learning for monocular depth estimation: A review. *Neurocomputing*, 438:14–33, May 2021. ISSN 0925-2312. doi: 10.1016/j.neucom.2020.12.089. URL http://dx.doi.org/10.1016/j.neucom.2020.12.089.

Mitra, D., Mallick, S., Paul, S., Chakrabarti, A., and Gupta, D. Exploring the benefits of ensemble techniques involving midas and monodepth2 models for monocular depth estimation. In *2024 4th International Conference on Computer, Communication, Control amp;amp; Information Technology (C3IT)*, pp. 1–6. IEEE, September 2024. doi: 10.1109/c3it60531.2024.10829443. URL http://dx.doi.org/10.1109/C3IT60531.2024.10829443.

Mukhoti, J. and Gal, Y. Evaluating bayesian deep learning methods for semantic segmentation, 2018. URL https://arxiv.org/abs/1811.12709.

Nix, D. and Weigend, A. Estimating the mean and variance of the target probability distribution. In *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)*, pp. 55–60 vol.1. IEEE, 1994. doi: 10.1109/icnn.1994.374138. URL http://dx.doi.org/10.1109/ICNN.1994.374138.

Obukhov, A., Poggi, M., Tosi, F., Arora, R. S., Spencer, J., Russell, C., Hadfield, S., Bowden, R., Wang, S., Ma, Z., Chen, W., Xu, B., Sun, F., Xie, D., Zhu, J., Lavreniuk, M., Guan, H., Wu, Q., Zeng, Y., Lu, C., Wang, H., Zhou, G., Zhang, H., Wang, J., Rao, Q., Wang, C., Liu, X., Lou, Z., Jiang, H., Chen, Y., Xu, R., Tan, M., Qin, Z., Mao, Y., Liu, J., Xu, J., Yang, Y., Zhao, W., Jiang, J., Liu, X., Zhao, M., Ming, A., Chen, W., Xue, F., Yu, M., Gao, S., Wang, X., Omotara, G., Farag, R., Demby, J., Tousi, S. M. A., DeSouza, G. N., Yang, T.-A., Nguyen, M.-Q., Tran, T.-P., Luginov, A., and Shahzad, M. The fourth monocular depth estimation challenge, 2025. URL https://arxiv.org/abs/2504.17787.

Ranftl, R., Bochkovskiy, A., and Koltun, V. Vision transformers for dense prediction, 2021. URL https://arxiv.org/abs/2103.13413.

Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation, 2015. URL https://arxiv.org/abs/1505.04597.

Shao, S., Li, R., Pei, Z., Liu, Z., Chen, W., Zhu, W., Wu, X., and Zhang, B. Towards comprehensive monocular depth estimation: Multiple heads are better than one. *IEEE Transactions on Multimedia*, 25:7660–7671, 2023. ISSN 1941-0077. doi: 10.1109/tmm.2022.3224810. URL http://dx.doi.org/10.1109/TMM.2022.3224810.

Zhang, C., Tian, K., Ni, B., Meng, G., Fan, B., Zhang, Z., and Pan, C. *Stereo Depth Estimation*

with Echoes, pp. 496–513. Springer Nature Switzerland, 2022. ISBN 9783031198120. doi: 10.1007/978-3-031-19812-0_29. URL http://dx.doi.org/10.1007/978-3-031-19812-0_29.

ŞAHiN, E., Arslan, N. N., and Özdemir, D. Unlocking the black box: an in-depth review on interpretability, explainability, and reliability in deep learning. *Neural Computing and Applications*, 37(2):859–965, November 2024. ISSN 1433-3058. doi: 10.1007/s00521-024-10437-2. URL http://dx.doi.org/10.1007/s00521-024-10437-2.

Appendix

A. Room-Type Expert Categories

The following table shows the breakdown of categories in terms of ResNet18 Places365 classifications. We selected the summarized subcategories of places365 to fit the theme so that all five categories are roughly equally sized. If we purely take a subcategory for an expert model, the number of images is too small, and the experts become too specialized, causing mode collapse (see Section 5.3.3). We split both training and validation images and obtained the following category counts.

Training set (19176 images):

- Sleeping: 4074
- Living: 2687
- Kitchen: 4547
- Work: 3667
- Remaining: 4201

Validation set (4795 images):

- Sleeping: 1025
- Living: 683
- Kitchen: 1099
- Work: 899
- Remaining: 1089

A. Sleeping

`bedroom, dorm_room, hotel_room, youth_hostel, childs_room, nursery, berth, alcove, bedchamber`

B. Living

`living_room, television_room, home_theater, recreation_room, playroom, bow_window/indoor, balcony/interior, movie_theater/indoor, music_studio, porch, patio, reception, corridor, lobby, entrance_hall, courtyard`

C. Kitchen

`kitchen, dining_room, pantry, galley, wet_bar, restaurant, coffee_shop, sushi_bar, restaurant_kitchen, ice_cream_parlor, bakery/shop, fastfood_restaurant, dining_hall, cafeteria`

D. Work

`home_office, office, office_cubicles, conference_room, waiting_room, computer_room, library/indoor, classroom, kindergarden_classroom, art_studio, lecture_room, chemistry_lab, physics_laboratory, biology_laboratory, art_school`

E. Remaining

`clothing_store, shoe_shop, hardware_store, beauty_salon, bookstore, department_store, drugstore, fabric_store, gift_shop, jewelry_shop, general_store/indoor, market/indoor, bazaar/indoor, toyshop, butchers_shop, pet_shop, bathroom, shower, closet, storage_room, basement, utility_room, laundromat, attic, garage/indoor, locker_room, dressing_room, sauna, jacuzzi/indoor, hospital_room, operating_room, veterinarians_office, clean_room, science_museum, natural_history_museum, museum/indoor, engine_room, server_room, repair_shop, elevator_lobby, elevator/door, staircase, atrium/public, television_studio, arena/rodeo, church/indoor, train_interior, bus_interior, elevator_shaft, jail_cell, burial_chamber, catacomb, aquarium, archive, phone_booth, bamboo_forest, throne_room, art_gallery, artists_loft, bank_vault, bowling_alley, gymnasium/indoor`

B. Image Augmentation

Another direction we explored was fine-tuning the base DPT model on augmented images.

Motivation. Early experiments showed that our DPT-based model struggles most with sharp edges (e.g., chair legs, shelf profiles), producing depth "bleed" around high-frequency boundaries.

Approach. To address this, we introduce two new expert models trained on augmented inputs:

Sharpened-image expert — we convolve each training and test image with the following 3×3 sharpening kernel to highlight edges and simplify their detection:

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Blurred-image expert — we convolve with the following separable 5×5 Gaussian kernel to soften edges, forcing the model to learn to recover them internally:

$$\begin{bmatrix} 0.00297 & 0.01331 & 0.02194 & 0.01331 & 0.00297 \\ 0.01331 & 0.05963 & 0.09832 & 0.05963 & 0.01331 \\ 0.02194 & 0.09832 & 0.16210 & 0.09832 & 0.02194 \\ 0.01331 & 0.05963 & 0.09832 & 0.05963 & 0.01331 \\ 0.00297 & 0.01331 & 0.02194 & 0.01331 & 0.00297 \end{bmatrix}$$

Both augmentations are applied *on-the-fly* during fine-tuning, as well as at inference time, and could be inter-swapped. The original idea behind this was to fine-tune the model on either 'easier' - sharpened images or 'harder' - smoothed ones, and test its performance on the images with the same augmentation, without any, and on the opposite augmentation (train on smoothed images, test on sharpened images). Both models were trained for 10 epochs on the full training dataset. As is shown in Figure 6, Figure 7, Figure 8, Figure 9, the smoothed model shows some improvement, which was aligned with lower siRMSE on the validation dataset, compared to the base model. We then tested the meta-model approach using the smoothed model as an expert. However, siRMSE score of this approach showed less improvement compared to room-type experts. It was decided not to pursue it further and switch to utilizing only room-type experts in the meta-model.

C. Uncertainty metrics

To estimate the quality of the produced uncertainty estimates we use the **PA**, **PU**, and **PAvPU** metrics proposed in (Mukhoti & Gal, 2018). First, we compute the maximum ratio r for each pixel using the Equation (5).

$$\max\left(\frac{y}{\hat{y}}, \frac{\hat{y}}{y}\right) = r. \tag{5}$$

Then, for each image we compute the median variance σ^2_{median} predicted by the model. We further define the following values: n_a – the number of accurately predicted pixels ($r < 1.25$), n_i – number of pixels where the model was inaccurate ($r \geq 1.25$), n_c – number of pixels in which the model is certain ($\sigma^2 < \sigma^2_{median}$) and n_u – number of pixels in which the model in uncertain ($\sigma^2 \geq \sigma^2_{median}$). We also define n_{ac} – number of accurate pixels in which the model is accurate and n_{iu} – number of uncertain pixels that are inaccurate. Finally, we compute the aforementioned metrics as follows:

- **PA** $= \mathbf{p}(\text{accurate}|\text{certain}) = \frac{n_{ac}}{n_c}$,
- **PU** $= \mathbf{p}(\text{uncertain}|\text{inaccurate}) = \frac{n_{iu}}{n_i}$,
- **PAvPU** $= \frac{n_{ac}+n_{ic}}{n_i+n_a}$.

D. Meta-model Combination Maps

D.1. Explanation of the Prediction Maps

In this section we examine how our meta-model combines the prediction maps of the individual experts. The illustrated images Figure 11 - Figure 15 are from the validation dataset.

In the first row, the original input RGB image is shown, followed by the depth prediction of the base model and the meta-model, respectively. The fourth image is the ground truth depth. The next image is the highest percentage λ_{ij} in the convex combination for each pixel, with the color (base model + expert models) illustrated in the legend. The last image in the upper row shows the absolute prediction error.

The second row illustrates the per-pixel percentage λ^k_{ij} for which a model $k \in I = [0, 1, .., K]$ is weighted. Moreover, we also depict the mean percentage over the image (the average λ_{ij}) and its standard deviation.

In the last row, we see the uncertainty of each respective expert.

The plots Figure 11, Figure 12, Figure 13, Figure 14, and Figure 15 show the predictions of the meta-model for a sample of each category. We also provide a sample that demonstrates a mode collapse inFigure 16, where the meta-model predominantly picks the base model if all experts are too weak.

We also provide an example, in which the convex combinations are relatively unstructured and therefore not well interpretable, which is the case when all experts are too similar in terms of performance (see Figure 17).

D.2. Discussion

We now examine the structure of the combined prediction maps by analyzing the meta-model's predictions of λ (probability) parameters to perform a sanity check on how the model perceives the image scene layout and whether it follows our intuition. For images with objects, we can observe that the relevant expert often receives the highest percentage and is preferred over the others for images with clear room types (like in a kitchen, for reference see Figure 13), although the transitions over surfaces are not as sharp as expected.

It is evident that the model combines the individual expert predictions, and the most strongly selected expert is often chosen within the boundary of an object surface. Furthermore, we can also see that the expert's uncertainty of the relevant category is the lowest.

If we try investigating which expert has the highest probability λ_{ij}, we notice that it is often used to predict the depth of less than half of the pixels in an image, indicating that the meta-model does not rely on one expert, but instead creates diverse combinations. However, these combinations are not always interpretable, as they sometimes lack structure. We mainly attribute to the experts not being specialized enough due to them sharing the same architecture. This could potentially be improved with a stronger meta-model architecture and stronger, more diverse expert models.

Layer	Type	Channels	Details
Input	-	3	RGB image
Enc1	Conv2d $\times 2$ + ReLU	$3 \to 64$	3×3, padding=1
Pool1	MaxPool2d	64	2×2
Enc2	Conv2d $\times 2$ + ReLU	$64 \to 128$	3×3, padding=1
Pool2	MaxPool2d	128	2×2
Bottleneck	Conv2d $\times 2$ + ReLU	$128 \to 256$	3×3, padding=1
Up2	Transposed Conv2d	$256 \to 128$	2×2, stride=2
Att2	Attention Block	128	Skip: Enc2
Dec2	Conv2d $\times 2$ + ReLU	$256 \to 128$	Concat(Up2, Enc2)
Up1	Transposed Conv2d	$128 \to 64$	2×2, stride=2
Att1	Attention Block	64	Skip: Enc1
Dec1	Conv2d $\times 2$ + ReLU	$128 \to 64$	Concat(Up1, Enc1)
Final	Conv2d	$64 \to N$	1×1 (output logits)

Table 3. Architecture of the Attention U-Net

Method	siRMSE	RMSE	MAE	REL	δ_1-Score	δ_2-Score	δ_3-Score
Our meta-model	0.0601	0.2380	0.1233	0.0429	0.9815	0.9965	0.9990
Base Model	0.0670	0.2802	0.1506	0.0527	0.9778	0.9958	0.9988
Averaging	0.0633	0.2522	0.1247	0.0421	0.9803	0.9960	0.9987
Inverse Variance-Weighted Averaging	0.0627	0.2479	0.1237	0.0420	0.9807	0.9962	0.9988
Category Oracle	0.0549	0.2254	0.1129	0.0394	0.9842	0.9974	0.9993
Optimistic Oracle	0.0405	0.1619	0.0200	0.0429	0.9907	0.9981	0.9994

Table 4. Evaluation metrics (columns) across all methods (rows), including our meta-model, baseline, expert combinations, and oracles.

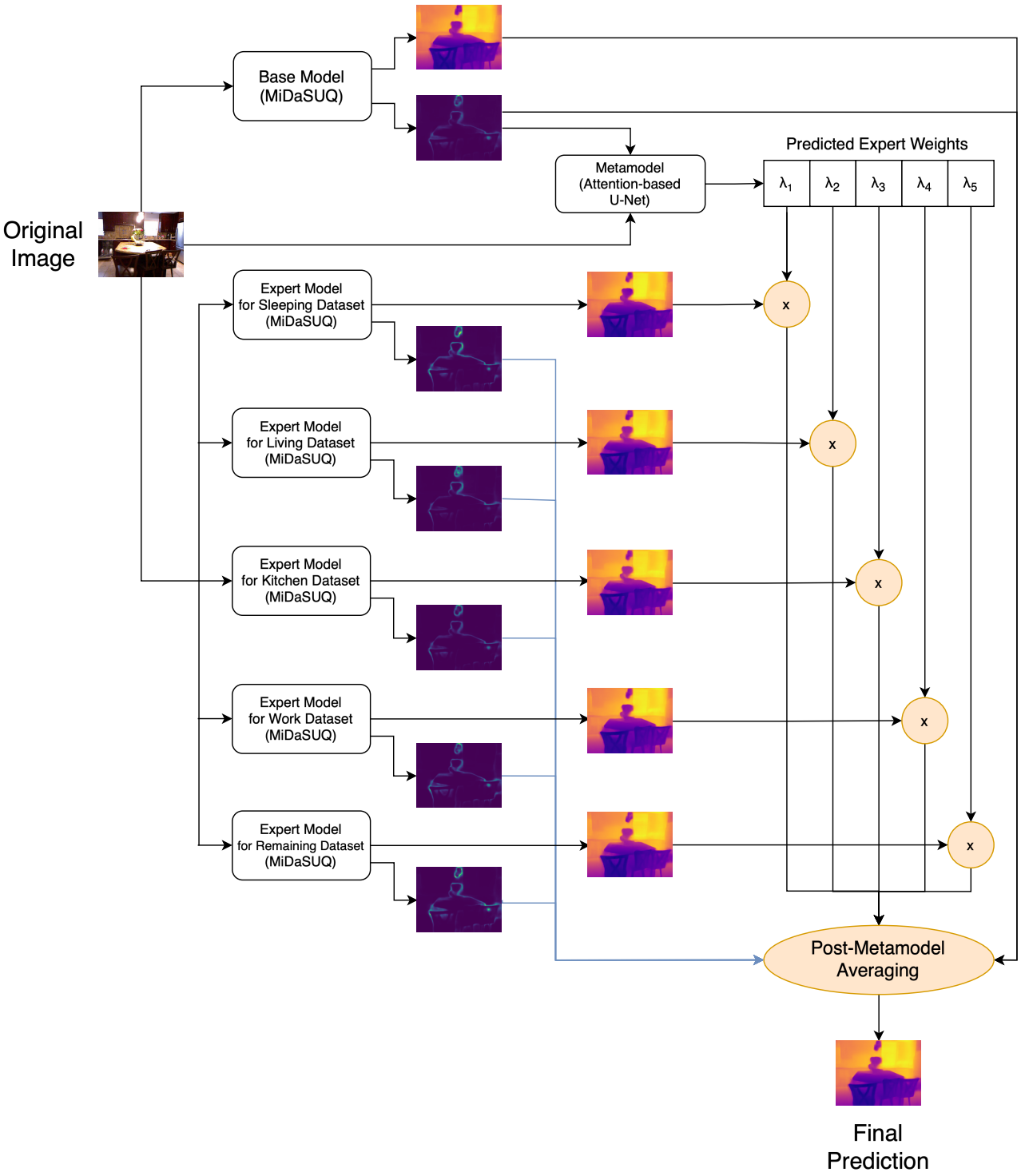

Figure 1. Illustration of how our pipeline works and how different models and components are combined to produce the final output.

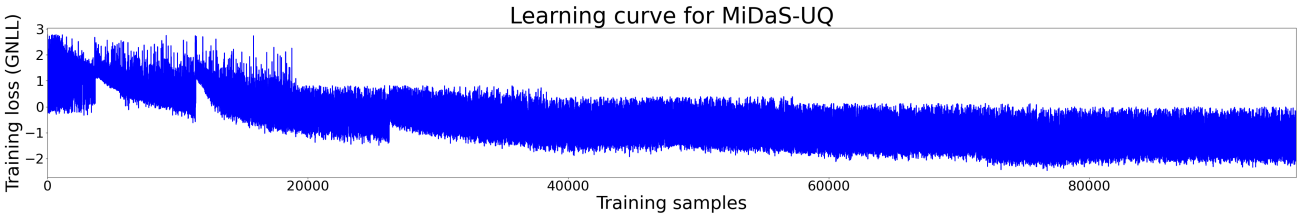

Figure 2. Base model learning curve. Training loss flattens out after 100k samples (5 epochs).

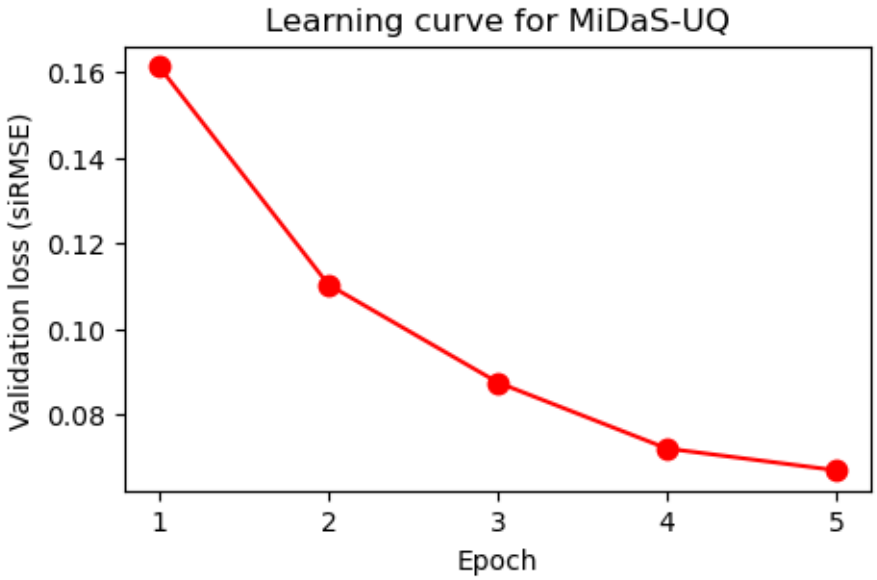

Figure 3. Base model learning curve. Validation loss is obtained at the end of each epoch.

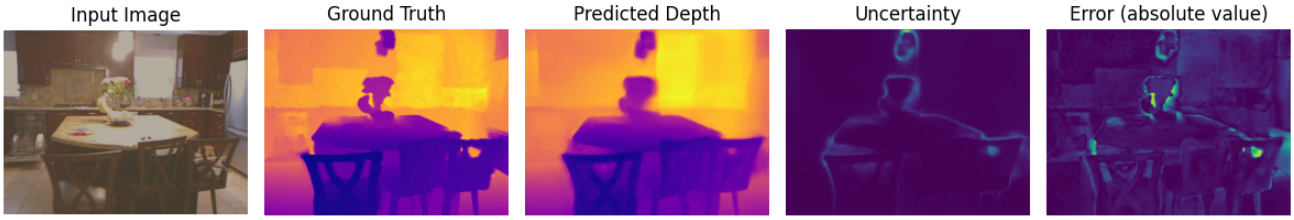

Figure 4. Example of base model prediction, uncertainty and error maps.

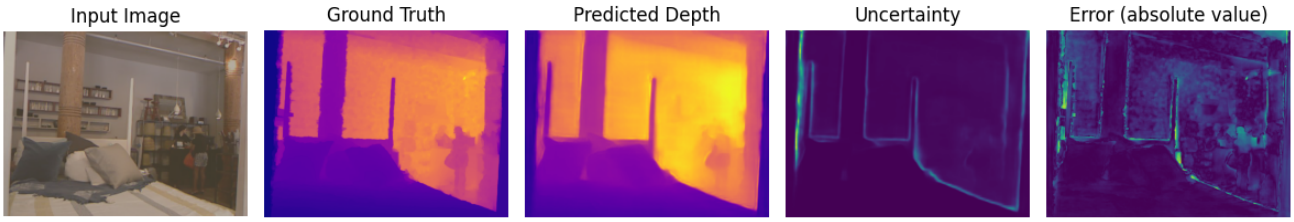

Figure 5. Example of base model prediction, uncertainty and error maps.

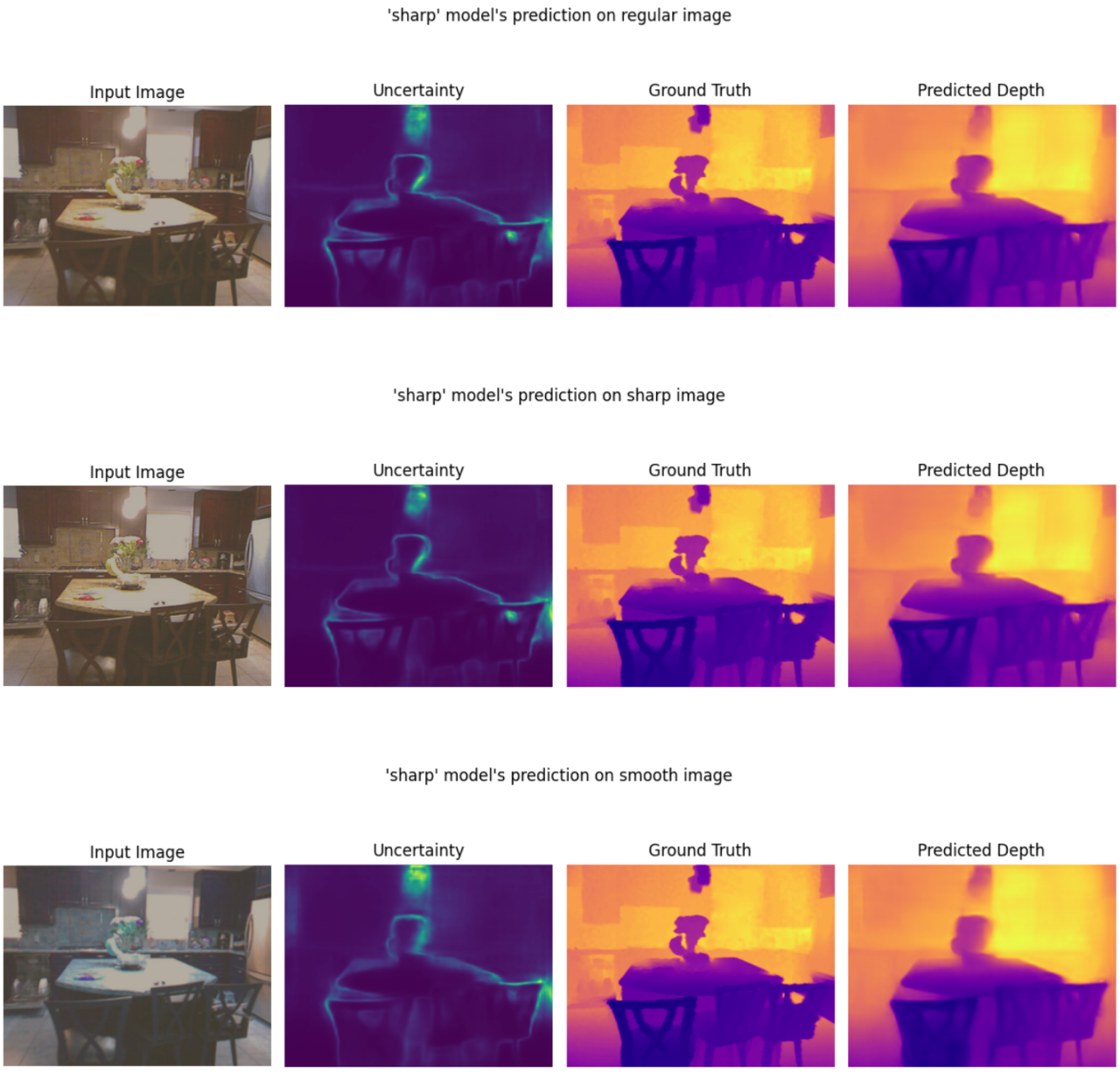

Figure 6. Comparison of 'sharp' model's inference and its uncertainty on image №1 with different augmentations

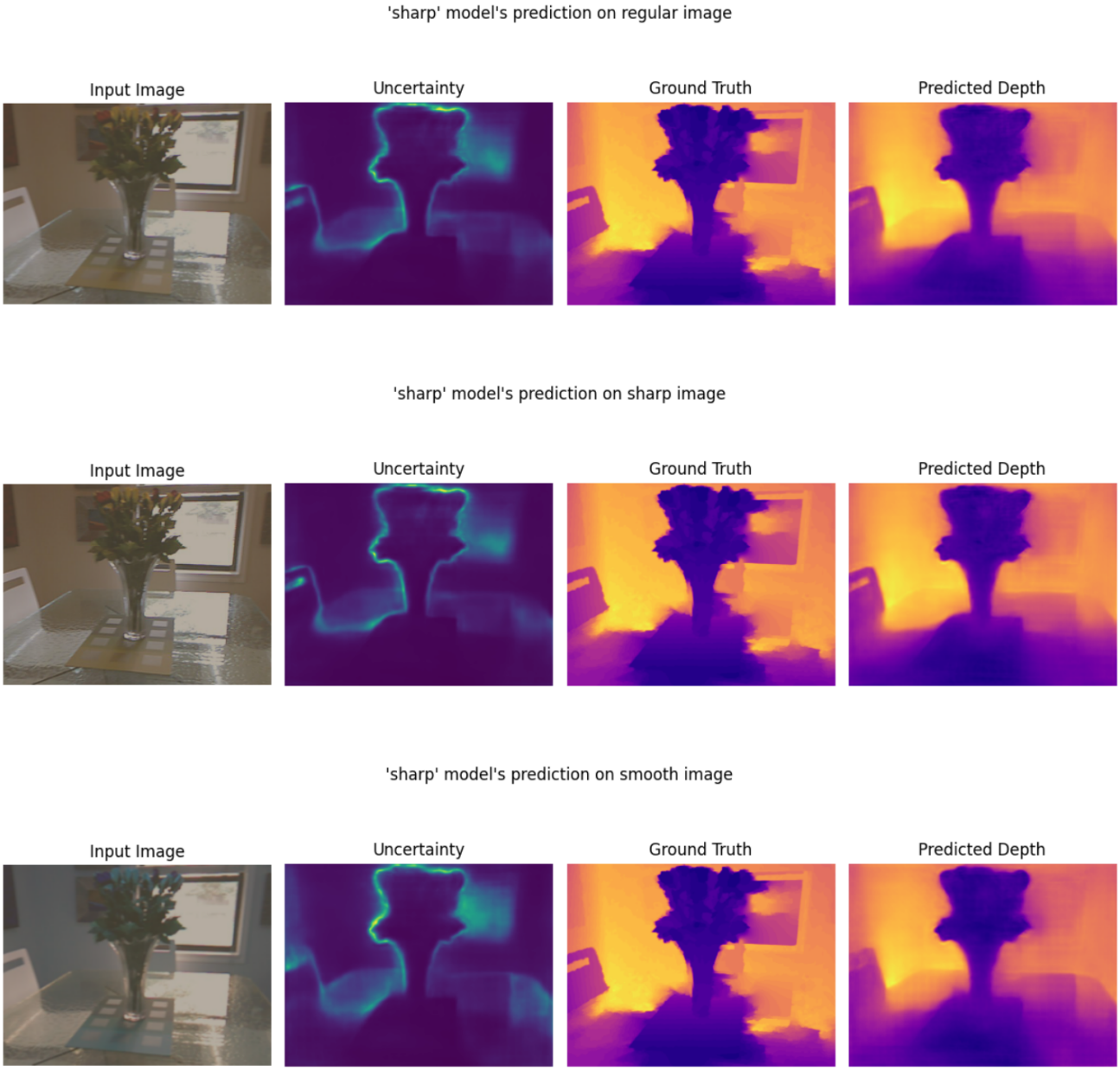

Figure 7. Comparison of 'sharp' model's inference and its uncertainty on image №2 with different augmentations

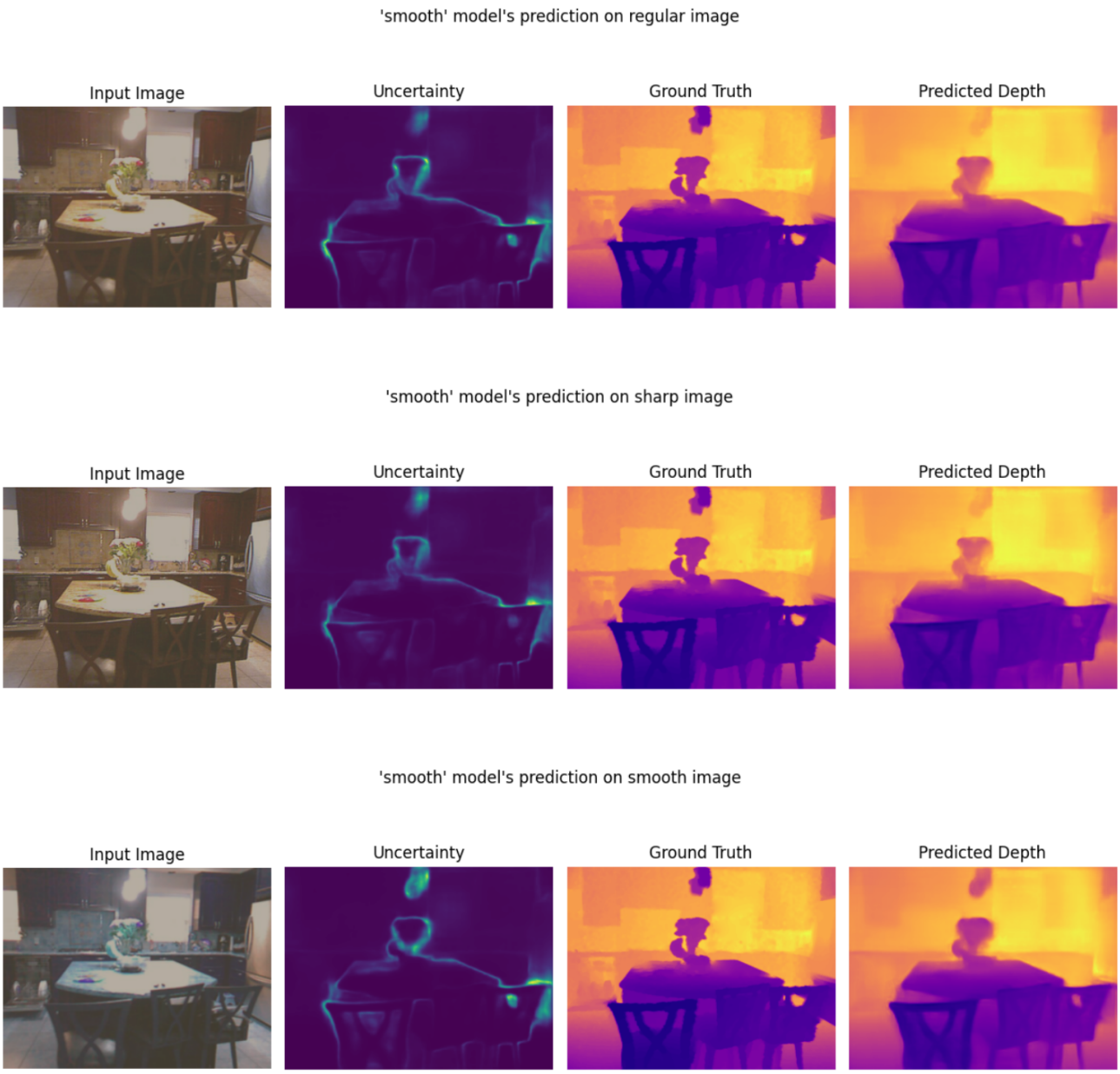

Figure 8. Comparison of 'smooth' model's inference and its uncertainty on image №1 with different augmentations

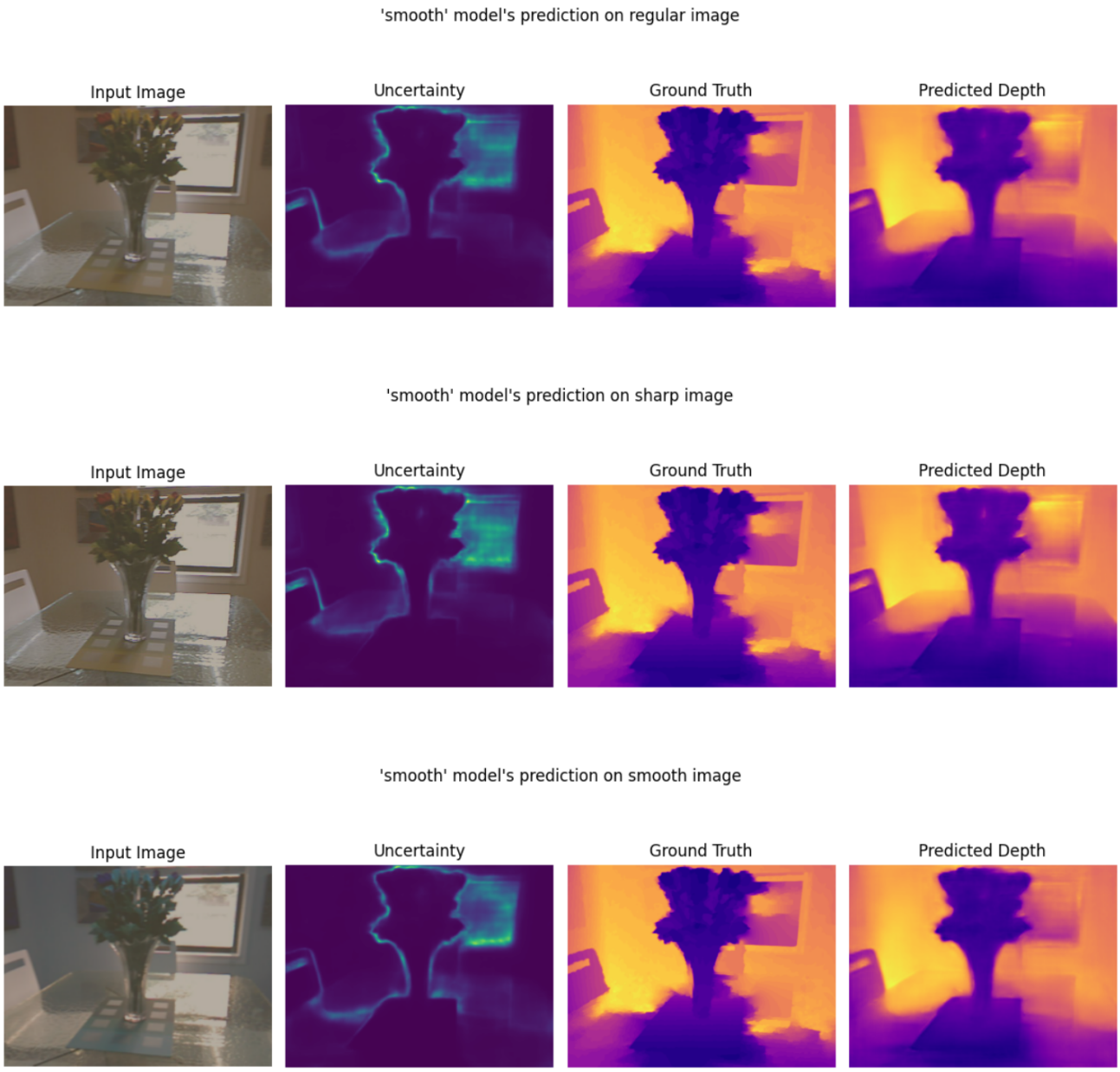

Figure 9. Comparison of 'smooth' model's inference and its uncertainty on image №2 with different augmentations

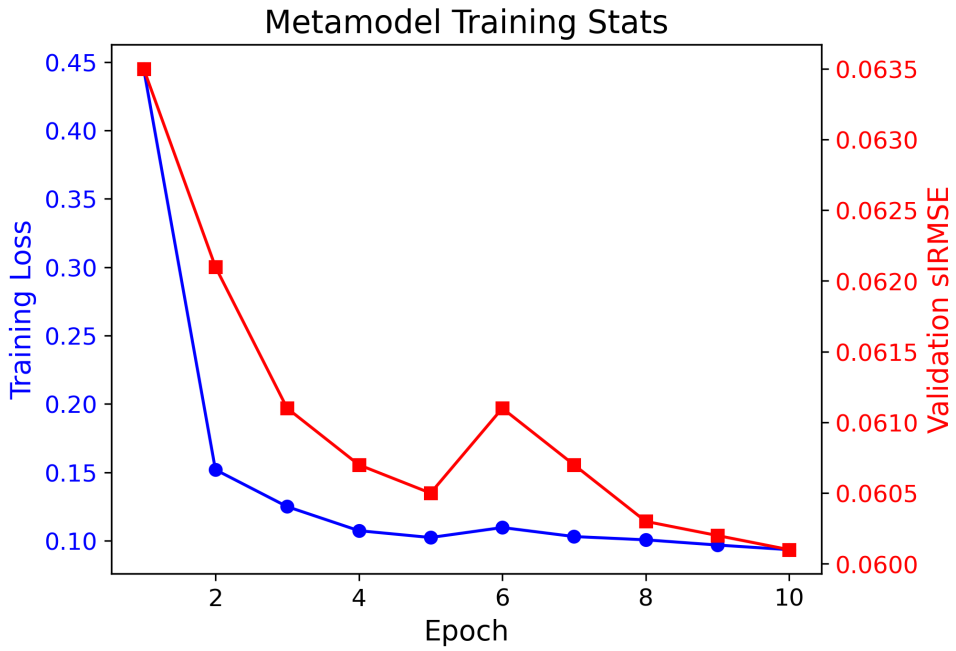

Figure 10. Meta-model training loss and validation siRMSE

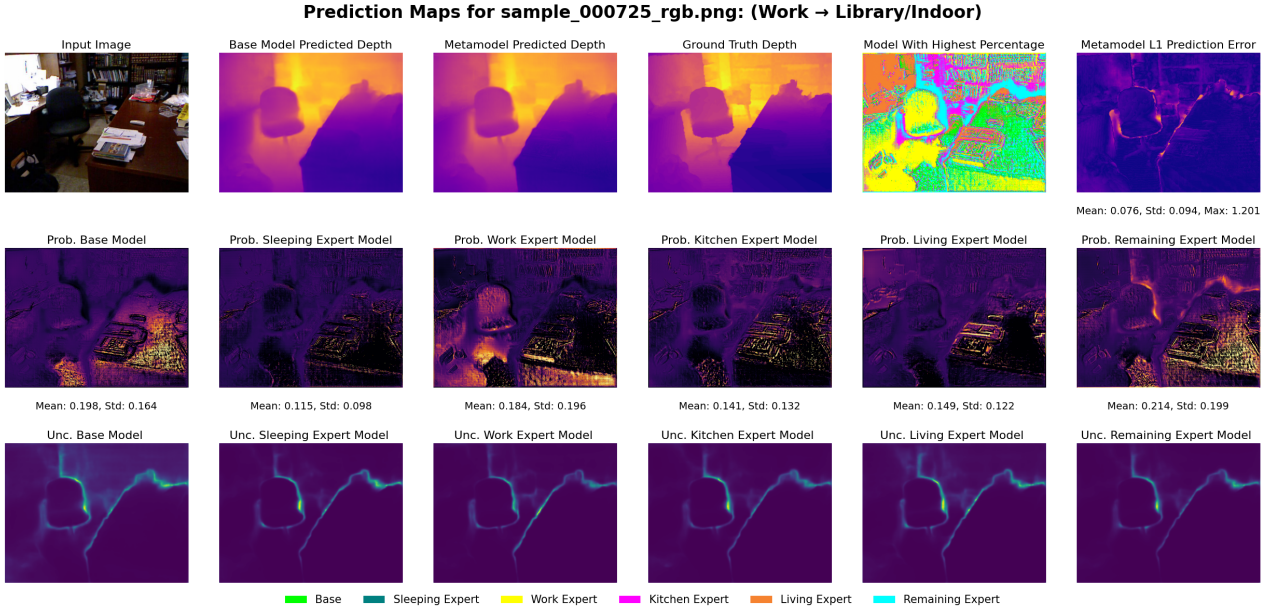

Figure 11. Meta-model ensemble prediction maps for a sample of ”Work”.

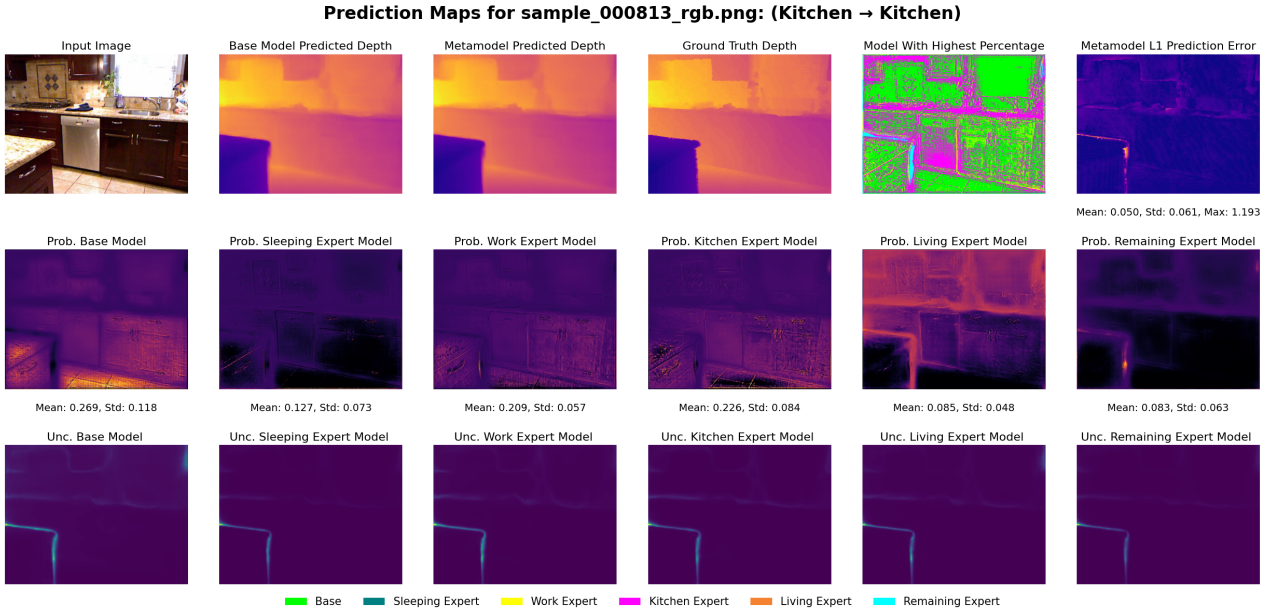

Figure 12. Meta-model ensemble prediction maps for a sample of ”Kitchen”.

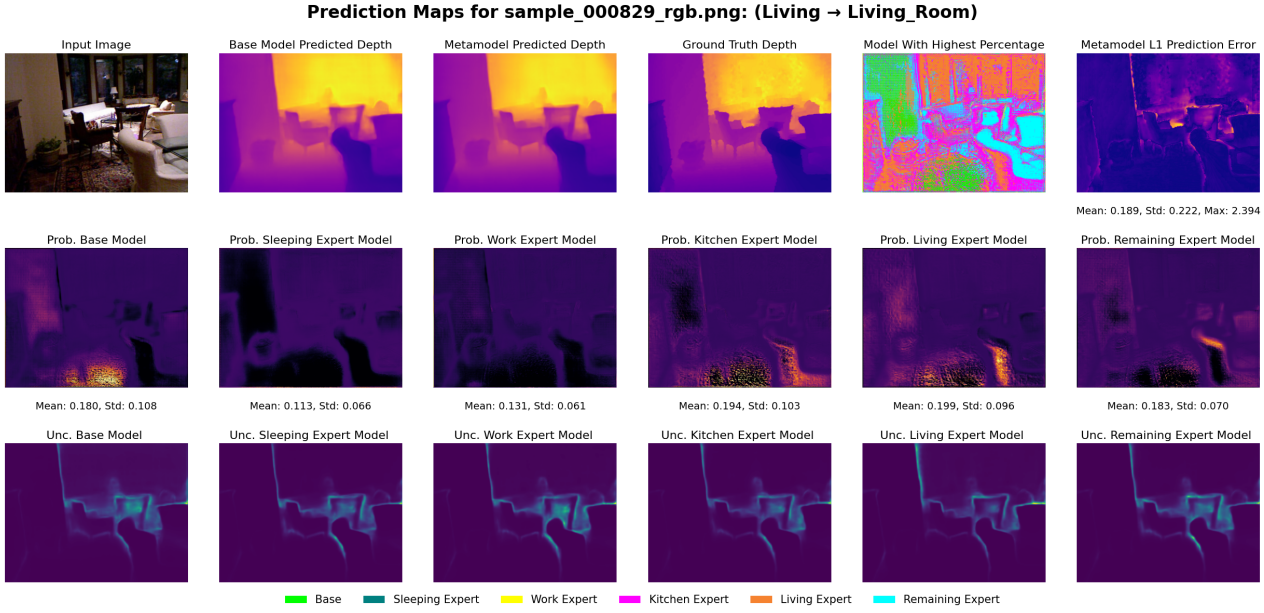

Figure 13. Meta-model ensemble prediction maps for a sample of ”Living”.

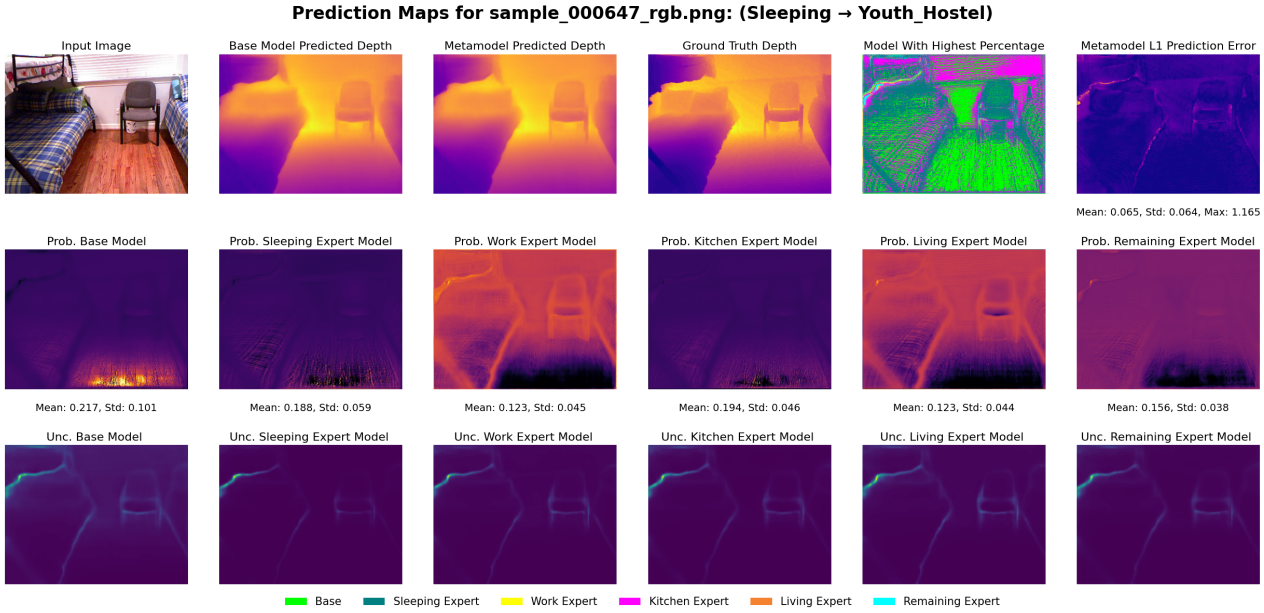

Figure 14. Meta-model ensemble prediction maps for a sample of ”Sleeping”.

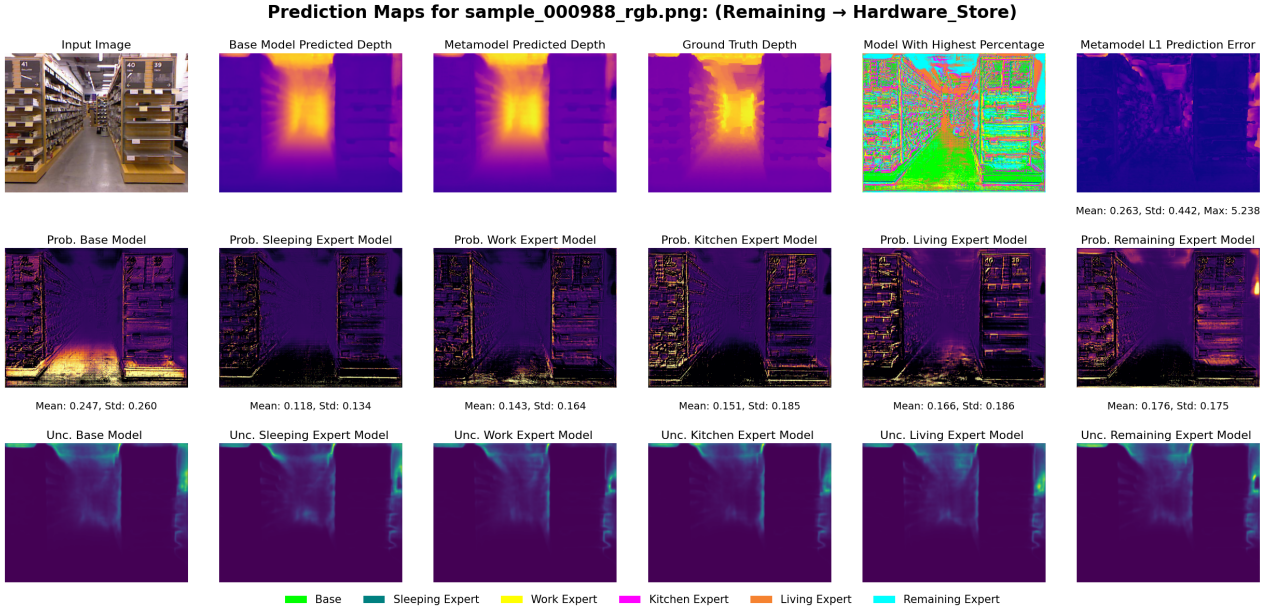

Figure 15. Matamodel ensemble prediction maps for a sample of ”Remaining”.

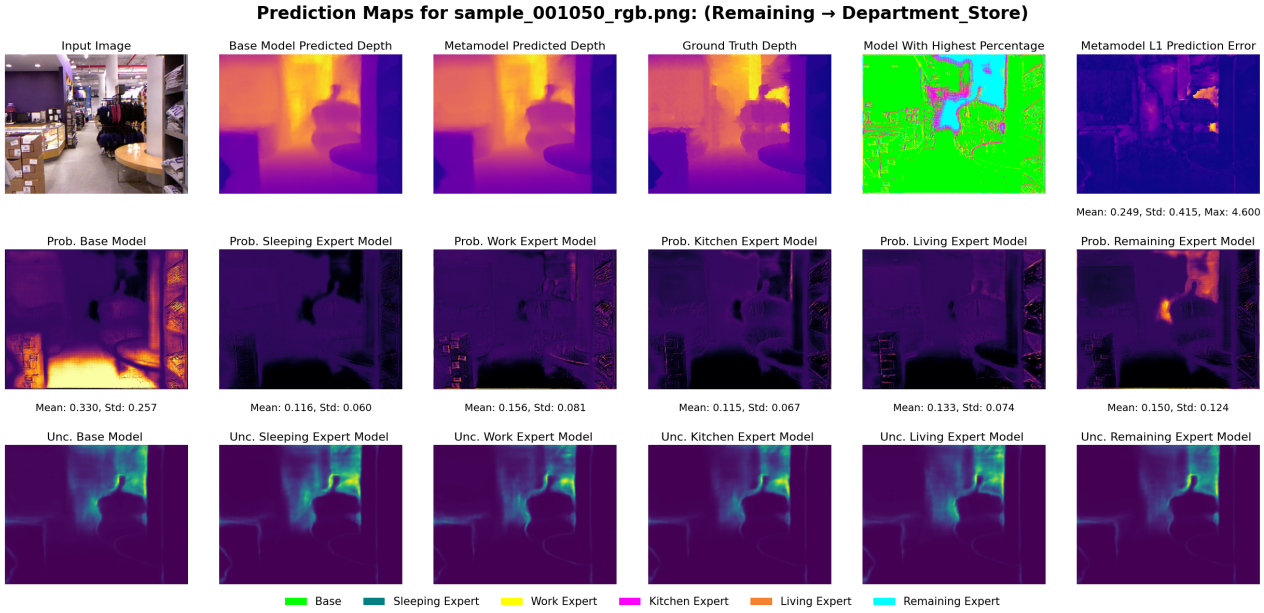

Figure 16. Meta-model Mode Collapse into Flat Distribution (experts are too weak)

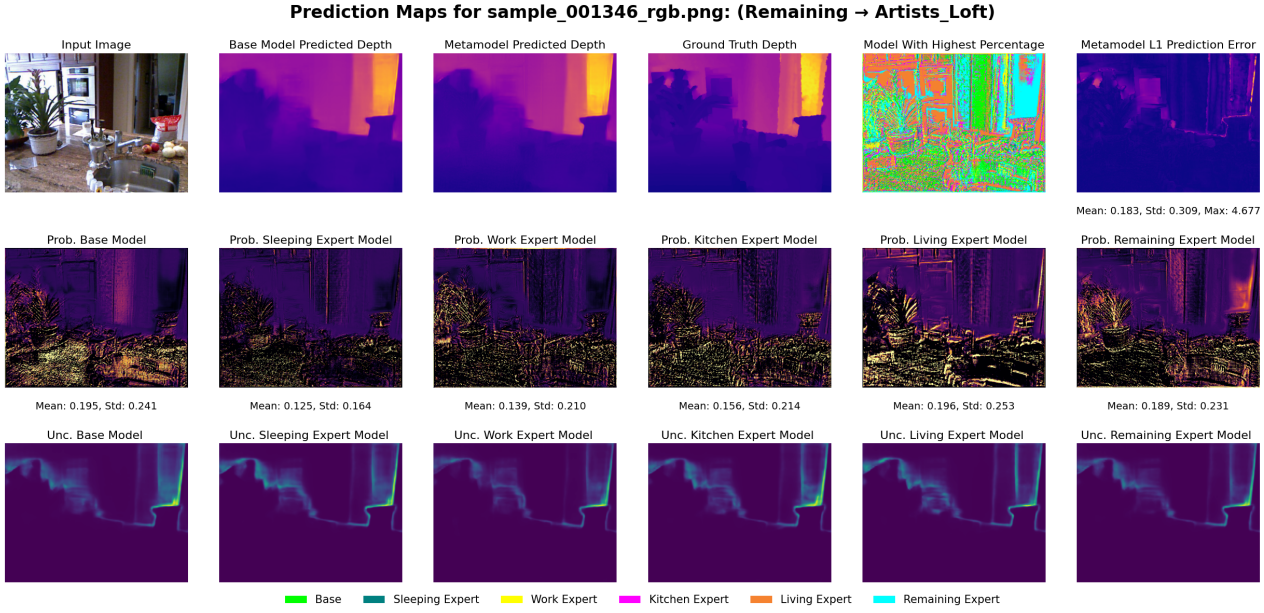

Figure 17. Meta-model combinations are flat (experts are too similar)