

Supplementary Materials: Unleashing Diffusion Models with Meta-Routers for Universal Few-Shot Dense Prediction

Anonymous Authors

1 DATASET DETAILS

We use the ‘tiny’ partition of Taskonomy dataset provided by [12], which consists of images and labels collected from 35 different buildings. To be consistent with VTM [5], we preprocess three single-channel tasks, namely Euclidean Distance (ED), Texture Edge (TE), and Occlusion Edge (OE), to enhance the task diversity:

- Texture edge (TE) labels are generated by applying Sobel edge detector to RGB images, which consists of a Gaussian filter and image gradient computation. The Gaussian filter has two hyper-parameters, namely kernel size and the standard deviation, where adjusting those hyper-parameters yield different thickness of detected edges. We use three different sets of hyper-parameters – (3, 1), (11, 2), (19, 3) – to produce 3-channel labels.
- Euclidean distance (ED) labels consists of pixel-wise depth map, where the depth is computed by the Euclidean distance from each image pixel to the camera’s optical center. As this task is very similar to the Z-buffer depth prediction (ZD) whose label pixels are the distance from each image pixel to the camera plane, we augment the ED task by segmenting the depth range and re-normalizing within each segment. Specifically, we compute the 5-quantiles of the pixel-wise depth labels in the whole dataset, then use each quantile as different channels after renormalization into [0, 1]. Thus the objective of each channel of the augmented ED task is to predict Euclidean distance within a specific range, where the ranges are disjoint for different channels.
- Occlusion edge (OE) labels are similar to texture edge, but they are constructed to depend on only the 3D geometry rather than color or lighting. We adopt the same approach as ED task to augment the OE labels into 5-channel labels.

The original labels for Semantic Segmentation (SS) task have the shape of $256 \times 256 \times 1$. Each position of the label is an integer value representing the ground truth semantic category. To convert these discrete labels into a continuous label structure, we follow VTM [5] to transform the labels into real values ranging from 0 to 1, resulting in the labels with the shape $\mathbb{R}^{256 \times 256 \times N_{cls}}$, where $N_{cls} = 12$ represents the total number of semantic classes. Within each channel, a value of 0 denotes the background region, while a value of 1 denotes the object region.

2 IMPLEMENTATION DETAILS

In this section, we elaborate on some implementation details that were not covered in the main body of the paper.

MoE transformation. To enable efficient adaptation through the use of router fine-tuning, we propose transforming the backbone into a MoE model. Specifically, we convert the multi-head attention layers and the feed-forward layers in the transformer blocks of the Stable Diffusion model’s denoising autoencoder ϵ_θ into MoE attention layers and MoE MLP layers. The weights of

attention experts $\{E_A^k\}$ in the MoE attention layer and MLP experts $\{E_F^k\}$ in the MoE MLP layer are initialized using the weights from the multi-head attention layer and the feed-forward layer in ϵ_θ . Below, we will present the details of this initialization process.

For the convenience of readers, we rewrite the definition of the MoE attention layer here. Given a query token $q^i \in \mathbb{R}^{1 \times C_M}$ from the query sequence Q , the objective of a MoE attention layer is to generate a new token y^i for q^i , with N_E^a attention experts $\{E_A^k\}_{k=1}^{N_E^a}$ and the router G_A :

$$y^i = \sum_{k=1}^{N_E^a} G_A^k(q^i) \cdot E_A^k(q^i), \quad (1)$$

$$E_A^k(q^i) = (\alpha^{i,k} V W_V) W_O^k, \quad (2)$$

$$\alpha^{i,k} = \text{Softmax}\left(\frac{q^i W_Q^k (K W_K)^T}{\sqrt{C_H}}\right), \quad (3)$$

where $W_K, W_V \in \mathbb{R}^{C_M \times C_H}$ are shared across attention experts to reduce computational complexity, while $W_Q^k \in \mathbb{R}^{C_M \times C_H}$ and $W_O^k \in \mathbb{R}^{C_H \times C_M}$ are specific to each expert. Here, C_M represents the dimension of input tokens and C_H represents the head dimension. G_A selects N_K^a experts and sets all other outputs to zero.

Next, we show how to initialize a MoE attention layer from a multi-head attention layer in ϵ_θ . Given the weights of the query linear layer $\hat{W}_Q \in \mathbb{R}^{h \times C_M \times C_H}$ and the output linear layer $\hat{W}_O \in \mathbb{R}^{h \times C_H \times C_M}$ from a multi-head attention layer in ϵ_θ , where h represents the number of attention heads, we firstly replicate the weights of \hat{W}_Q and \hat{W}_O by b times, obtaining $\tilde{W}_Q \in \mathbb{R}^{hb \times C_M \times C_H}$ and $\tilde{W}_O \in \mathbb{R}^{hb \times C_H \times C_M}$. Then, we initialize a MoE attention layer’s weights of $W_Q \in \mathbb{R}^{N_E^a \times C_M \times C_H}$ and $W_O \in \mathbb{R}^{N_E^a \times C_H \times C_M}$ from the weights of \tilde{W}_Q and \tilde{W}_O , where $N_E^a = hb$. The weights of $W_K, W_V \in \mathbb{R}^{C_M \times C_H}$ are initialized randomly. To maintain the computational complexity between the MoE attention layer and the multi-head attention layer in ϵ_θ , the router G_A selects the same number of attention heads as the multi-head attention layer. This is achieved by setting N_K^a as h , which is 8 in the Stable Diffusion model. In our experiments, we set b as 3. Therefore, we use 24 experts with top- k as 8 for MoE attention layers.

As for a feed-forward layer in ϵ_θ , the Stable Diffusion model adopts a variant of Gated Linear Units (GLU) called GEGLU [8]:

$$\mathcal{F}_{GEGLU}(x) = (\Phi(x \hat{W}_G) \odot x \hat{W}_I) \hat{W}_O, \quad (4)$$

where Φ represents the GELU activation function [3]. $\Phi(x \hat{W}_G)$ act as gating values. $\hat{W}_G, \hat{W}_I \in \mathbb{R}^{C_M \times C_D}$ and $\hat{W}_O \in \mathbb{R}^{C_D \times C_M}$, where $C_D = \hat{h} \times C_M$, $\hat{h} \in \mathbb{N}$. To transform \mathcal{F}_{GEGLU} into MoE MLP layer, we split $\hat{W}_G, \hat{W}_I, \hat{W}_O$ into \hat{h} parts and replicate them by b times, resulting in $\{\hat{W}_G^k\}_{k=1}^{hb}, \{\hat{W}_I^k\}_{k=1}^{hb}, \{\hat{W}_O^k\}_{k=1}^{hb}$, where $\hat{W}_G^k, \hat{W}_I^k, \hat{W}_O^k \in$

$\mathbb{R}^{C_M \times C_M}$. Then, a MoE MLP layer can be represented as:

$$y = \sum_{k=1}^{N_E^f} G_F^k(x) \cdot E_F^k(x), \quad (5)$$

$$E_F^k(x) = (\Phi(x\hat{W}_G^k) \odot x\hat{W}_F^k)\hat{W}_O^k, \quad (6)$$

where $N_E^f = \hat{h}b$. G_F retains the outputs of N_K^f experts and sets all other outputs to zero. To maintain the computation complexity before and after the MoE transformation, we set N_K^f as \hat{h} , which is 4 in the Stable Diffusion model. In our experiments, we set b as 3. Therefore, we use 12 experts with top-k as 4 for MoE MLP layers.

Details for extracting task embedding. We enhance the meta-routers' capabilities by incorporating the task embedding extracted from the support samples. To extract the task embedding, we introduce a task embedding extractor ξ based on the concatenated input of the image and the label. This approach has been proven effective in representing vision tasks [10]. As the dimensions of labels can vary across different tasks, we transform the label $Y^i \in \mathbb{R}^{H \times W \times C_T}$ for each sample (X^i, Y^i) of task \mathcal{T} into $\tilde{Y}^i \in \mathbb{R}^{H \times W \times 3}$, which is then concatenated with the input image $X^i \in \mathbb{R}^{H \times W \times 3}$. Here, we delve into the specific details of this transformation process. First, we define three sets of parameters $\{a^j, b^j\}_{j=1}^3$. Then, for j th set of parameters $\{a^j, b^j\}$, we apply the linear projection $a^j x + b^j$ for each channel of Y^i and average the results of all channels, obtaining $\bar{Y}^i \in \mathbb{R}^{H \times W}$. Finally, we stack the results obtained from these three sets of parameters $\{a^j, b^j\}_{j=1}^3$ into $\tilde{Y}^i \in \mathbb{R}^{H \times W \times 3}$.

Task-specific convolution layer. The task-specific convolution layers used in the prediction head are 3×3 convolution layers with the input dimension of 128 and output dimension of C_T for each task \mathcal{T} , which are randomly initialized for different task in the multi-task pre-training stage. However, in the subsequent stages, to enhance the model's adaptation ability, we initialize all these task-specific convolution layers from a single 3×3 convolution layer with the input dimension of 128 and output dimension of 1. This design allows all these task-specific convolution layers to share a set of initialization weights which can be optimized in the meta-training stage to enable rapid adaptation to novel few-shot tasks.

3 DETAILS OF STABLE DIFFUSION MODEL

To begin with, we provide a brief overview of the Denoising Diffusion Probabilistic Model (DDPM) [4], upon which the Stable Diffusion model[6] is built. DDPM transforms the noise $z_T \sim \mathcal{N}(0, I)$ to the sample z_0 by gradually denoising z_T to less noisy samples, which is the reverse process of a diffusion process. Formally, a diffusion process is modeled as a Markov:

$$z_t \sim \mathcal{N}(\sqrt{\alpha_t}z_{t-1}, (1 - \alpha_t)I), \quad (7)$$

where $\{\alpha_t\}$ are fixed coefficients that determine the noise schedule. A noisy sample z_t can be obtained directly from the data z_0 :

$$z_t = \sqrt{\alpha_t}z_0 + \sqrt{1 - \alpha_t}\epsilon, \quad \epsilon \sim \mathcal{N}(0, 1), \quad (8)$$

where $\bar{\alpha}_t = \prod_{s=1}^t \alpha_s$. This further allows us to sample an arbitrary z_t efficiently during training. The training objective of diffusion models can be derived as [4]:

$$\mathcal{L}_{DM} = \mathbb{E}_{z_0, \epsilon, t} \|\epsilon - \epsilon_\theta(z_t(z_0, \epsilon), t; C)\|^2, \quad (9)$$

Table 1: Ablation study of the MoE attention layer's expert number N_E^a and the MoE MLP layer's expert number N_E^f , where $N_E^a = hb$ and $N_E^f = \hat{h}b$, with $h = 8$ and $\hat{h} = 4$. $b \in \mathbb{N}$ is the replication factor, which is set as 3 in our experiments.

b	N_E^a	N_E^f	SS (mIoU \uparrow)	SN (mErr \downarrow)
1	8	4	0.3829	11.1813
2	16	8	0.4057	9.8769
3	24	12	0.4310	9.1261
4	32	16	0.4334	9.2053

Table 2: Comparison of different initialization methods for the MoE backbone of our method.

Method	SS (mIoU \uparrow)	SN (mErr \downarrow)
Random Init.	0.3935	10.7280
Partial Random Init.	0.4169	9.7856
Replication Init.	0.4310	9.1261

where z_t is computed as Equation (8). ϵ_θ is a denoising autoencoder (usually implemented as a UNet [7]) that is learned to predict the ϵ given the conditioning input C , which can be the text prompt. The sampling of diffusion models is achieved by discretizing the diffusion SDE or ODE [9], which requires multiple model evaluations at different timesteps.

Recently, a new type of diffusion model called latent diffusion model [6] is proposed. Stable Diffusion is a latent diffusion model trained on large-scale image-text dataset LAION-5B, which has demonstrated remarkable performance on image synthesis controlled by natural language. Specifically, the Stable Diffusion Model first train a VQGAN model [2], which comprises an encoder \mathcal{E} and a decoder \mathcal{D} , enabling conversion between the pixel space and the latent space. Subsequently, a diffusion model is trained on this latent space with the same objective in Equation (9). Although diffusion models are trained using generative loss, their features have shown impressive performance in specific visual perception tasks that demand a comprehensive understanding of pixel-level fine-grained information [1, 11, 13], such as semantic segmentation and depth estimation. In this work, we take a step towards exploiting the Stable Diffusion model's rich features for few-shot dense prediction tasks in a universal manner. We focus on leveraging the pre-trained knowledge from Stable Diffusion by treating the UNet-like denoising autoencoder ϵ_θ as a vision backbone, without considering the language condition and the diffusion process.

4 ADDITIONAL RESULTS

In this section, we conduct ablation experiments on the expert number of MoE attention layers and MoE MLP layers. We also compare several initialization methods for our MoE backbone to leverage the pre-trained knowledge of Stable Diffusion.

Ablation on expert number. As discussed in Section 2, we adopt an initialization strategy for the weights of the MoE attention layers and MoE MLP layers in the MoE backbone, which involves replicating the weights of the multi-head attention layers and feed-forward layers of Stable Diffusion by a factor of b and initialize the MoE backbone from these weights. Consequently, the expert number N_E^a of the MoE attention layer is equal to hb , while the expert

number N_E^f of the MoE MLP layer is equal to $\hat{h}b$, where $h = 8$ and $\hat{h} = 4$. In both cases, the replication factor b determines the overall number of experts. Therefore, we conduct ablation experiments on the replication factor b , ranging from 1 to 4, as shown in Table 1. As observed, increasing b from 1 to 3 consistently improves the performance. However, there is little improvement when b increases from 3 to 4. Hence, we choose b as 3 to strike a balance between performance and memory cost.

Comparing initialization methods for MoE backbone. To demonstrate the effectiveness of our initialization strategy, we compare several initialization methods in Table 2. ‘Random Init.’ represents randomly initializing the weights of all experts within the MoE backbone. ‘Partial Random Init.’ represents initializing a subset of h attention experts in the MoE attention layer and \hat{h} MLP experts in the MoE MLP layer from the weights of the Stable Diffusion model’s multi-head attention layer and feed-forward layer, while randomly initializing the remaining experts. ‘Replication Init.’ is the initialization strategy adopted by our method, which involves replicating the weights from Stable Diffusion by a factor of b and utilizing them to initialize the MoE backbone. As observed in the table, our initialization method proves to be the most effective approach for enabling the MoE backbone to leverage the pre-trained knowledge of the Stable Diffusion model.

REFERENCES

- [1] Dmitry Baranchuk, Andrey Voynov, Ivan Rubachev, Valentin Khulkov, and Artem Babenko. 2022. Label-Efficient Semantic Segmentation with Diffusion Models. In *International Conference on Learning Representations*.
- [2] Patrick Esser, Robin Rombach, and Bjorn Ommer. 2021. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 12873–12883.
- [3] Dan Hendrycks and Kevin Gimpel. 2017. Bridging Nonlinearities and Stochastic Regularizers with Gaussian Error Linear Units. (2017).
- [4] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems* 33 (2020), 6840–6851.
- [5] Donggyun Kim, Jinwoo Kim, Seongwoong Cho, Chong Luo, and Seunghoon Hong. 2023. Universal Few-shot Learning of Dense Prediction Tasks with Visual Token Matching. In *The Eleventh International Conference on Learning Representations*.
- [6] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10684–10695.
- [7] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention—MICCAI 2015: 18th international conference, Munich, Germany, October 5–9, 2015, proceedings, part III* 18. Springer, 234–241.
- [8] Noam Shazeer. 2020. Glue variants improve transformer. *arXiv preprint arXiv:2002.05202* (2020).
- [9] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. 2021. Score-Based Generative Modeling through Stochastic Differential Equations. In *International Conference on Learning Representations*.
- [10] Xinlong Wang, Wen Wang, Yue Cao, Chunhua Shen, and Tiejun Huang. 2023. Images speak in images: A generalist painter for in-context visual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6830–6839.
- [11] Jiarui Xu, Sifei Liu, Arash Vahdat, Wonmin Byeon, Xiaolong Wang, and Shalini De Mello. 2023. Open-vocabulary panoptic segmentation with text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2955–2966.
- [12] Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. 2018. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3712–3722.
- [13] Wenliang Zhao, Yongming Rao, Zuyan Liu, Benlin Liu, Jie Zhou, and Jiwen Lu. 2023. Unleashing text-to-image diffusion models for visual perception. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 5729–5739.