RepMode: Learning to Re-parameterize Diverse Experts for Subcellular Structure Prediction

Donghao Zhou 1,2,3 Chunbin Gu 3 Junde Xu 1,2,3 Furui Liu 4 Qiong Wang 1
Guangyong Chen 4* Pheng-Ann Heng 1,3,4

1 Guangdong Provincial Key Laboratory of Computer Vision and Virtual Reality Technology,
Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences
2 University of Chinese Academy of Sciences
3 The Chinese University of Hong Kong
4 Zhejiang Lab

https://correr-zhou.github.io/RepMode

Abstract

In biological research, fluorescence staining is a key technique to reveal the locations and morphology of subcellular structures. However, it is slow, expensive, and harmful to cells. In this paper, we model it as a deep learning task termed subcellular structure prediction (SSP), aiming to predict the 3D fluorescent images of multiple subcellular structures from a 3D transmitted-light image. Unfortunately, due to the limitations of current biotechnology, each image is partially labeled in SSP. Besides, naturally, subcellular structures vary considerably in size, which causes the multi-scale issue of SSP. To overcome these challenges, we propose Re-parameterizing Mixture-of-Diverse-Experts (RepMode), a network that dynamically organizes its parameters with task-aware priors to handle specified single-label prediction tasks. In RepMode, the Mixture-of-Diverse-Experts (MoDE) block is designed to learn the generalized parameters for all tasks, and gating re-parameterization (GatRep) is performed to generate the specialized parameters for each task, by which RepMode can maintain a compact practical topology exactly like a plain network, and meanwhile achieves a powerful theoretical topology. Comprehensive experiments show that RepMode can achieve state-of-the-art overall performance in SSP.

1. Introduction

Recent years have witnessed great progress in biological research at the subcellular level [1,2,7,21,23,59,61], which plays a pivotal role in deeply studying cell functions and behaviors. To address the difficulty of observing subcellular structures, fluorescence staining was invented and has become a mainstay technology for revealing the locations and morphology of subcellular structures [26]. Specifically, biologists use the antibodies coupled to different fluorescent dyes to “stain” cells, after which the subcellular structures of interest can be visualized by capturing distinct fluorescent signals [64]. Unfortunately, fluorescence staining is expensive and time-consuming due to the need for advanced instrumentation and material preparation [29]. Besides, phototoxicity during fluorescent imaging is detrimental to living cells [28]. In this paper, we model fluorescence

*Corresponding Author: gychen@zhejianglab.com
staining as a deep learning task, termed subcellular structure prediction (SSP), which aims to directly predict the 3D fluorescent images of multiple subcellular structures from a 3D transmitted-light image (see Fig. 1(a)). The adoption of SSP can significantly reduce the expenditure on subcellular research and free biologists from this demanding workflow.

Such an under-explored and challenging bioimage problem deserves the attention of the computer vision community due to its high potential in biology. Specifically, SSP is a dense regression task where the fluorescent intensities of multiple subcellular structures need to be predicted for each transmitted-light voxel. However, due to the limitations of current biotechnology, each image can only obtain partial labels. For instance, some images may only have the annotations of nucleoli, and others may only have the annotations of microtubules (see Fig. 1(b)). Moreover, different subcellular structures would be presented at multiple scales under the microscope, which also needs to be taken into account. For example, the mitochondrion is a small structure inside a cell, while obviously the cell membrane is a larger one since it surrounds a cell (see Fig. 1(c)).

Generally, there are two mainstream solutions: 1) Multi-Net [5,32,33,47]: divide SSP into several individual prediction tasks and employs multiple networks; 2) Multi-Head [6,9,46]: design a partially-shared network composed of a shared feature extractor and multiple task-specific heads (see Fig. 2(a)). However, these traditional approaches organize network parameters in an inefficient and inflexible manner, which leads to two major issues. First, they fail to make full use of partially labeled data in SSP, resulting in label-inefficiency. In Multi-Net, only the images containing corresponding labels would be selected as the training set for each network and thus the other images are wasted, leading to an unsatisfactory generalization ability. As for Multi-Head, although all images are adopted for training, only partial heads are updated when a partially labeled image is input and the other heads do not get involved in training. Second, to deal with the multi-scale nature of SSP, they require exhausting pre-design of the network architecture, and the resultant one may not be suitable for all subcellular structures, which leads to scale-inflexibility.

In response to the above issues, herein we propose Reparameterizing Mixture-of-Diverse-Experts (RepMode), an all-shared network that can dynamically organize its parameters with task-aware priors to perform specified single-label prediction tasks of SSP (see Fig. 2(b)). Specifically, RepMode is mainly constructed of the proposed Mixture-of-Diverse-Experts (MoDE) blocks. The MoDE block contains the expert pairs of various receptive fields, where these task-agnostic experts with diverse configurations are designed to learn the generalized parameters for all tasks. Moreover, gating re-parameterization (GatRep) is proposed to conduct the task-specific combinations of experts to achieve efficient expert utilization, which aims to generate the specialized parameters for each task. With such a parameter organizing manner (see Fig. 2(c)), RepMode can maintain a practical topology exactly like a plain network, and meanwhile achieves a theoretical topology with a better representational capacity. Compared to the above solutions, RepMode can fully learn from all training data, since the experts are shared with all tasks and thus participate in the training of each partially labeled image. Besides, RepMode can adaptively learn the preference of each task for the experts with different receptive fields, thus no manual intervention is required to handle the multi-scale issue. Moreover, by fine-tuning few newly-introduced parameters, RepMode can be easily extended to an unseen task without any degradation of the performance on the previous tasks. Our main contributions are summarized as follows:

- We propose a stronger baseline for SSP, named Rep-Mode, which can switch different “modes” to predict multiple subcellular structures and also shows its potential in task-incremental learning.
- The MoDE block is designed to enrich the generalized parameters and GatRep is adopted to yield the specialized parameters, by which RepMode achieves dynamic parameter organizing in a task-specific manner.
- Comprehensive experiments show that RepMode can achieve state-of-the-art (SOTA) performance in SSP. Moreover, detailed ablation studies and further analysis verify the effectiveness of RepMode.

2. Related Works

Partially labeled dense prediction. In addition to SSP, many other dense prediction tasks could also face the challenge of partial labeling. In general, the previous methods can be divided into two groups. The first one seeks for an effective training scheme by adopting knowledge distillation [17,72], learning cross-task consistency [36], designing jointly-optimized losses [54], etc. The second one aims to improve the network architecture with a dynamic segmentation head [70], task-guided attention modules [62], conditional tensor incorporation [15], etc. However, these methods are primarily developed for large-scale datasets. Compared to these well-explored tasks, SSP only has relatively small datasets due to the laborious procedure of fluorescence staining. Thus, the training data of SSP should be utilized in a more efficient way. In light of that, we adopt a task-conditioning strategy in RepMode, where all parameters are shared and thus can be directly updated using the supervision signal of each label. Unlike other task-conditional networks [15,55,62,70], our RepMode is more flexible and capable of maintaining a compact topology.

Multi-scale feature learning. Multi-scale is a fundamental problem of computer vision, caused by the variety
in the size of the objects of interest. The common solutions are adopting multi-resolution input [16, 19, 66, 71], designing parallel branches [3, 25, 37, 39, 57, 74], fusing cross-layer features [42, 44, 50, 56], performing hierarchical predictions [40, 41, 43], etc. These methods often adopt a predefined architecture for all objects with different receptive fields for each subcellular structure, and thus is capable of learning multi-scale features in a unified fashion. In contrast, RepMode learns the dynamic combinations of the experts with different receptive fields for each subcellular structure, and thus is capable of learning multi-scale features in a task-specific manner.

Mixture-of-Experts. Mixture-of-Experts (MoE) typically consists of a gating module and multiple independent learners (i.e., experts) [67]. For an input sample, MoE would adaptively assemble the corresponding output of all experts [10, 45, 48, 58] or only route it to a few specific experts [24, 31, 51, 53], which depends on the gating strategy. Benefiting from its divide-and-conquer principle, MoE is widely adopted in computer vision [10, 20, 48, 51, 63], natural language processing [8, 22, 53], and recommendation systems [24, 45, 49, 58]. Our RepMode is established based on the idea of MoE, but is further explored from the following aspects: 1) Instead of performing input-aware gating, RepMode only uses the task embedding for gating, aiming to adjust its behavior for a specified task; 2) The experts of RepMode can be combined together, which can efficiently utilize multiple experts in an MoE-inspired architecture.

Structural re-parameterization. Different from other re-parameterization (re-param) methods [18, 35, 52, 68], structural re-param [13, 14] is a recent technique of equivalently converting multi-branch network structures. With this technique, multi-branch blocks [12–14, 60] are introduced to plain networks for enhancing their performance. However, these methods only achieve inference-time converting, resulting in non-negligible training costs. There are previous works [11, 27] accomplishing training-time converting, but they require model-specific optimizer modification [11] or extra parameters [27] and only explore its potential on one single task. In this work, we elegantly incorporate task-specific gating into structural re-param to achieve both training- and inference-time converting for handling multiple tasks, which is more cost-friendly and with better applicability. Besides, dynamic convolutions [4, 38, 65, 73] also can be roughly considered as re-param methods, which aim to assemble convolutions with the same shape in an input-dependent way. In contrast, using task-dependent gating, our RepMode can combine experts with diverse configurations to generate composite convolutional kernels, and thus is with higher flexibility to model more situations.

3. Methodology

3.1. Problem definition

We start by giving a formal definition of SSP. Following [47], we assume that each image has only one fluorescent label, which greatly relaxes the annotation requirement and makes the setting of this task more general and challenging. Let \( D = \{ (x_n, y_n, l_n) \}_{n=1}^{N} \) denotes a SSP dataset with \( N \) samples. The \( n \)-th image \( x_n \in \mathcal{I}_n \) is associated with the label \( y_n \in \mathcal{I}_n \), where \( \mathcal{I}_n = \mathbb{R}^{D_n \times H_n \times W_n} \) denotes the image space and \( D_n \times H_n \times W_n \) is the image size. The label indicator \( l_n \in \mathcal{L} = \{1, 2, ..., S\} \) represents that \( y_n \) is the label of the \( l_n \)-th subcellular structure, where \( S \) is the total number of subcellular structure categories. In this work, our goal is to learn a network \( F : \mathcal{I} \times \mathcal{L} \rightarrow \mathcal{I} \) with the parameters \( \theta \) from \( \mathcal{D} \). SSP can be considered as a collection of \( S \) single-label prediction tasks, each of which corresponds to one category of subcellular structures. To solve SSP, Multi-Net and Multi-Head divide task-specific parameters from \( \theta \) for each task. In contrast, RepMode aims to share \( \theta \) with all tasks and dynamically organize \( \theta \) to handle specified tasks.
3.2. Network architecture

The backbone of RepMode is a 3D U-shape encoder-decoder architecture mainly constructed of the downsampling and upsampling blocks. Specifically, each downsampling block contains two successive MoDE blocks to extract task-specific feature maps and double their channel number, followed by a downsampling layer adopting a convolution with a kernel size of $2 \times 2 \times 2$ and a stride of 2 to halve their resolution. Note that batch normalization (BN) and ReLU activation are performed after each convolutional layer. In each upsampling block, an upsampling layer adopts a transposed convolution with a kernel size of $2 \times 2 \times 2$ and a stride of 2 to upsample feature maps and halve their channel number. Then, the upsampled feature maps are concatenated with the corresponding feature maps passed from the encoder, and the resultant feature maps are further refined by two successive MoDE blocks. Finally, a MoDE block without BN and ReLU is employed to reduce the channel number to 1, aiming to produce the final prediction. We adopt such a common architecture to highlight the applicability of RepMode and more details are provided in Appendix A. Notably, MoDE blocks are employed in both the encoder and decoder, which can facilitate task-specific feature learning and thus helps to achieve superior performance.

3.3. Mixture-of-Diverse-Experts block

To handle various prediction tasks of SSP, the representational capacity of the network should be strengthened to guarantee its generalization ability. Thus, we propose the MoDE block, a powerful alternative to the vanilla convolutional layer, to serve as the basic network component of RepMode. In the MoDE block, diverse experts are designed to explore a unique convolution collocation, and the gating module is designed to utilize the task-aware prior to produce gating weights for dynamic parameter organizing. We delve into the details of these two parts in the following.

Diverse expert design. In the MoDE block, we aim to achieve two types of expert diversity: 1) Shape diversity: To tackle the multi-scale issue, the experts need to be equipped with various receptive fields; 2) Kernel diversity: Instead of irregularly arranging convolutions, it is a better choice to explore a simple and effective pattern to further enrich kernel combinations. Given these guidelines, we propose to construct expert pairs to constitute the multi-branch topology of the MoDE block. The components of an expert pair are 3D convolutions (Conv) and 3D average poolings (Avgp). Specifically, an expert pair contains a Conv $K \times K \times K$ expert and an Avgp $K \times K \times K$ - Conv $1 \times 1 \times 1$ expert, and we utilize a stride of 1 and same-padding to maintain the resolution of feature maps. Overall, the MoDE block is composed of expert pairs with three receptive fields to attain shape diversity (see Fig. 3(a)). When $K = 1$, since these two experts are equal, only one is preserved for simplicity.

Notably, the Avgp $K \times K \times K$ - Conv $1 \times 1 \times 1$ expert is essentially a special form of the Conv $K \times K \times K$ expert. To be specific, merging the serial Avgp $K \times K \times K$ kernel and Conv $1 \times 1 \times 1$ kernel would result in a Conv kernel with limited degrees of freedom (named as A-Conv). Compared to normal Conv, A-Conv has only one learnable parameter and thus acts like a learnable average pooling, which enriches kernel diversity in the same shape (see Fig. 3(b)).

The combination of Conv and Avgp is also widely adopted in previous works [13, 27], but we further explore such a characteristic from the perspective of serial merging.

Gating module design. In order to perform a specified single-label prediction task, the task-aware prior needs to be encoded into the network, so that it can be aware of which task is being handled and adjust its behavior to focus on the desired task. Instead of embedding the task-aware prior by a hash function [15] or a complicated learnable module [55], we choose the most simple way, i.e. embed the task-aware prior of each input image $x_n$ with the label indicator $l_n$ into a S-dimensional one-hot vector $p_n$, which is expressed as

$$p_{n,s} = \begin{cases} 1, & \text{if } s = l_n, \\ 0, & \text{otherwise}, \end{cases} \quad s = 1, 2, \ldots, S, \quad (1)$$

where $p_{n,s}$ indicates the $s$-th entry of $p_n$. Then, the task embedding $p_n$ is fed into the gating module and the gating weights $G$ are generated by a single-layer fully connected network (FCN) $\phi(\cdot)$, shown as $G = \phi(p_n) = \{g_t\}_{t=1}^T$ where $T = 5$. Note that we omit $n$ in $G$ for brevity. Here $g_t \in \mathbb{R}^{C_O}$ represents the gating weights for the $t$-th expert, which is split from $G$, and $C_O$ is the channel number of the output feature maps. Finally, $G$ would be further activated as $\hat{G} = \{\hat{g}_t\}_{t=1}^T$ by Softmax for the balance of the intensity of different experts, which can be formulated as

$$\hat{g}_{ti} = \frac{\exp(g_{ti})}{\sum_{j=1}^T \exp(g_{ji})}, \quad i = 1, 2, \ldots, C_O, \quad (2)$$

where $g_{ti}$ (resp. $\hat{g}_{ti}$) is the $i$-th entry of $g_t$ (resp. $\hat{g}_t$). With the resultant gating weights $\hat{G}$, RepMode can perform dynamic parameter organizing for these task-agnostic experts conditioned on the task-aware prior.
3.4. Gating re-parameterization

In addition to studying expert configurations, how to efficiently utilize multiple experts is also worth further exploration. The traditional manner is to completely utilize all experts to process the input feature maps [10,45,48,58] (see Fig. 4(a)). However, the output of all experts needs to be calculated and stored, which would slow down training and inference and increase the GPU memory utilization [14,69]. The advanced one is to sparsely route the input feature maps to specific experts [24,31,51,53] (see Fig. 4(b)). However, only a few experts are utilized and the others remain unused, which would inevitably reduce the representational capacity of MoE. To avoid these undesired drawbacks and meanwhile preserve the benefits of MoE, we elegantly introduce task-specific gating to structural re-param, and thus propose GatRep to adaptively fuse the kernels of experts in the MoDE block, through which only one convolution operation is explicitly required (see Fig. 4(c)).

**Preliminary.** GatRep is implemented based on the homogeneity and additivity of Conv and Avgp, which are recognized in [13,14]. The kernels of a Conv with $C_1$ input channels, $C_0$ output channels, and $K \times K \times K$ kernel size is a fifth-order tensor $W \in \mathcal{Z}(K) = \mathbb{R}^{C_0 \times C_1 \times K \times K \times K}$, where $\mathcal{Z}(K)$ denotes this kernel space. Besides, the kernels of an Avgp $W^a \in \mathbb{R}^{C_1 \times C_1 \times K \times K \times K}$ can be constructed by

$$W^a_{c_1,c_1',...}: = \begin{cases} \frac{1}{K^3}, & \text{if } c_1 = c_1', \\ 0, & \text{otherwise,} \end{cases}$$

where $c_1,c_1',...$ is the indexes of the tensor and $c_1, c_1' = 1, ..., C_1$. Note that the Avgp kernel is fixed and thus unlearnable. Moreover, we omit the biases here as a common practice. Specifically, GatRep can be divided into two steps, i.e. serial merging and parallel merging (see Fig. 5).

**Step 1: serial merging.** The first step of GatRep is to merge Avgp and Conv into an integrated kernel. For brevity, here we take an Avgp - Conv expert as an example. Let $M^I$ denote the input feature maps. The process of producing the output feature maps $M^O$ can be formulated as

$$M^O = W \odot (W^a \odot M^I),$$

where $\odot$ denotes the convolution operation. According to the associative law, we can perform an equivalent transformation for Eq. (4) by first combining $W^a$ and $W$. Such a transformation can be expressed as

$$M^O = \left( W \odot W^a \right) \odot M^I,$$

which means that we first adopt $W$ to perform a convolution operation on $W^a$, and then use the resultant kernel $W^c$ to process the input feature maps. With this transformation, the kernels of Avgp and Conv can be merged as an integrated one for the subsequent step.

**Step 2: parallel merging.** The second step of GatRep is to merge all experts in a task-specific manner. We define $\text{Pad}(\cdot, K')$ as a mapping function equivalently transferring a kernel to the kernel space $\mathcal{Z}(K')$ by zero-padding, and set $K' = 5$ which is the biggest receptive field size of these experts. Let $M^O$ denote the final task-specific feature maps. This transformation can be formulated as

$$M^O = \left( \sum_{t=1}^{T} \hat{g}_t \odot \text{Pad}(W^e_t, K') \right) \odot M^I,$$

where $\odot$ denotes the channel-wise multiplication and $W^e_t$ is the kernel of the $t$-th expert. Note that $W^e_t$ is an integrated kernel (resp. Conv kernel) for an Avgp - Conv expert (resp. Conv expert). To be specific, the detailed pixel-level form is provided in Appendix B. Finally, $W^c$ is the resultant task-specific kernel dynamically generated by GatRep.

4. Experiments

4.1. Experimental setup

We conduct the experiments based on the following experimental setup unless otherwise specified. Due to space limitations, more details are included in Appendix C.
Datasets. For a comprehensive comparison, the dataset is constructed from a dataset collection [47] containing twelve partially labeled datasets, each of which corresponds to one category of subcellular structures (i.e. one single-label prediction task). All images are resized to make each voxel correspond to a size of 0.29 × 0.29 × 0.29 mm³. Moreover, we perform per-image z-scored normalization for voxels to eliminate systematic differences and thus is less sensitive to outliers than MSE.

Implementation details. Mean Squared Error (MSE) is adopted as the loss function, which is commonly used to train a regression model. Besides, Adam [34] is employed as the optimizer with a learning rate of 0.0001. Each model is trained for 1000 epochs from scratch and validation is performed every 20 epochs. Finally, the validated model that attains the lowest MSE is selected for evaluation on the test set. In a training epoch, we randomly crop a patch with size of 32 × 128 × 128 from each training image as the input with a batch size of 8, and random flip is performed for data augmentation. In the inference stage, we adopt the Gaussian sliding window strategy [30] to aggregate patch-based output for a full prediction. To ensure fairness, the same backbone architecture, training configuration, and inference strategy are applied to all comparing models.

Evaluation metrics. In addition to MSE, Mean Absolute Error (MAE) and Coefficient of Determination ($R^2$) are also used as the evaluation metrics. MAE measures absolute differences and thus is less sensitive to outliers than MSE. $R^2$ measures correlations by calculating the proportion of variance in a label that can be explained by its prediction. For a clear comparison, we also present the relative overall performance improvement over Multi-Net (i.e. $\Delta{\text{Imp}}$).

4.2. Comparing to state-of-the-art methods

We compared our RepMode to the following methods: 1) Multi-Net: [47]; 2) Multi-Head: include two variants, i.e. multiple task-specific decoders (denoted by Dec.) or last layers (denoted by Las.); 3) CondNet [15] and TSNs [55]: two SOTA task-conditional networks for multi-task learning; 4) PIPO-FAN [16], DoDNet [70], and TGNNet [62]: three SOTA methods of a similar task, i.e. partially labeled multi-organ and tumor segmentation (note that DoDNet and TGNNet also adopt task-conditioning strategies).

The experimental results on twelve tasks of SSP are reported in Tab. 1. As recognized in [47], the performance of Multi-Net is sufficient to assist with biological research in some cases, thus it can be a reference of reliable metric values for real-life use. Furthermore, two Multi-Head variants can achieve better performance, which verifies the importance of learning from the complete dataset. Notably, PIPO-FAN is an improved Multi-Head variant that additionally constructs a pyramid architecture to handle the multi-scale issue. The results show that such an architecture can further improve performance but still can not address this issue well. Moreover, the competitive performance of CondNet and TSNs demonstrates that adopting an appropriate task-conditioning strategy is beneficial. However, these
networks remain Multi-Head variants since multiple task-specific heads are still required. As an advanced version of DoDNet, TGNet additionally modulates the feature maps in the encoder and skip connections, leading to more competitive performance. It can be observed that, SSP is an extremely tough task since it is hard to attain a huge performance leap in SSP, even for those powerful SOTA methods of related tasks. However, the proposed RepMode, which aims to learn the task-specific combinations of diverse task-agnostic experts, outperforms the existing methods on ten of twelve tasks of SSP and achieves SOTA overall performance. Notably, RepMode can even achieve 7.209% (resp. 4.482%, 9.176%) imp on MSE (resp. MAE, $R^2$), which is near twice the second best method (i.e. TGNet).

### 4.3. Ablation studies

To verify the effectiveness of the proposed RepMode, we conduct comprehensive ablation studies totally including four aspects, where the results are reported in Tab. 2.

#### Scope of use of the MoDE block

As we mentioned in Sec. 3.2, we employ the MoDE block in both the encoder and decoder of the network. Therefore, we change its scope of use to explore its influence on performance. The results show that employing it only in the encoder can achieve better performance than only in the decoder, since the encoder can extract task-specific features and pass them to the decoder through skip connections. Moreover, employing it in both the encoder and decoder is superior since the whole network can perform dynamic parameter organizing.

#### Effectiveness of the expert design

The MoDE block is composed of three expert pairs, each of which contains a Conv expert and an Avgp - Conv expert. It can be observed that removing experts (especially the Conv experts) from the MoDE block could cause a performance drop due to the degradation of the representational capacity. Moreover, it is also an interesting finding that the expert pair with the commonly used $3 \times 3 \times 3$ receptive field is most critical.

#### Average poolings matter

Avgp is one of the basic components of the MoDE block. The results show that removing such an unlearnable component could also reduce performance, which further verifies the effectiveness of the expert design. Besides, we can observe that setting the receptive fields of all Avgp to the same one could also cause a performance drop. This is because being equipped with different receptive fields could facilitate expert diversity.

#### Different gating strategies

For task-specific gating, the one-hot task embedding is fed into the single-layer FCN followed by Softmax activation. Accordingly, we conduct the following modifications: 1) Use the task embedding with each entry sampled from $\mathcal{N}(0,1)$; 2) Use the two-layer FCN with the hidden unit number set to 6; 3) Use Sigmoid activation; 4) Input-dependent gating: input feature maps are first processed by a global average pooling and then fed into the gating module. The superior performance and simplicity of the original gating approach demonstrate the applicability of RepMode. Notably, input-dependent gating underperforms since an all-shared network can not be aware of the desired task of input without access to any priors.

### 4.4. Further analysis

In this subsection, we perform further analysis of RepMode to further reveal its capability. Additional analysis and discussion are provided in Appendix D.

#### Gating weights visualization

In the MoDE block, the gating weights are produced for dynamic parameter organizing, through which the preference of each task for diverse experts can be learned. As shown in Fig. 6, the cell membrane relatively prefers the Conv $5 \times 5 \times 5$ expert while the mitochondrion relatively prefers the Conv $1 \times 1 \times 1$ one as we expect. Besides, the preference of the mid-scale structures (i.e. nucleolus and nuclear envelope) is more variable. Notably, the Avg - Conv experts also be assigned sufficient weights, which verifies the effectiveness of the expert pairs.
It can also be observed that the network pays more attention to small-scale features in the decoder, which could be due to the need for producing a detailed prediction.

**Qualitative results.** Subcellular structures are hard to be distinguished in transmitted-light images (see Fig. 7). The second best method (i.e., TGNet) suffers from incomplete (see Fig. 7(a)) and redundant (see Fig. 7(b)) predictions, and even yields inexistent patterns (see Fig. 7(c)). But relatively, RepMode can produce more precise predictions for various subcellular structures at multiple scales even though there are some hard cases (see Fig. 7(a)&(b)). Such a practical advance is crucial in biological research, since inaccurate predictions at some key locations may mislead biologists into making incorrect judgments.

**RepMode as a better task-incremental learner.** For a well-trained RepMode, we fine-tuned a newly-introduced expert and gating module for each MoDE block with the other experts frozen, aiming to extend it to an unseen task (see Appendix D.1 for more details). As we expect, RepMode can preserve and transfer its domain knowledge through the pretrained experts, which helps to achieve better performance compared to the plain networks (see Tab. 3).

As long as the previous gating weights have been stored, such a task-incremental learning manner would not cause any degradation of the performance on the previous tasks.

### 5. Conclusions

In this paper, we focus on an under-explored and challenging bioimage problem termed SSP, which faces two main challenges, i.e., partial labeling and multi-scale. Instead of constructing a network in a traditional manner, we choose to dynamically organize network parameters with task-aware priors and thus propose RepMode. Experiments show that RepMode can achieve SOTA performance in SSP. We believe that RepMode can serve as a stronger baseline for SSP and help to motivate more advances in both the biological and computer vision community.

### Acknowledgements

We would like to thank Danruo Deng, Bowen Wang, and Jiancheng Huang for their valuable discussion and suggestions. This work is supported by the National Key R&D Program of China (2022YFE0200700), the National Natural Science Foundation of China (Project No. 62006219 and No. 62072452), the Natural Science Foundation of Guangdong Province (2022A1515011579), the Regional Joint Fund of Guangdong under Grant 2021B1515120011, and the Hong Kong Innovation and Technology Fund (Project No. ITS/170/20 and ITS/241/21).
References


[48] Zhen Qin, Yicheng Cheng, Zhe Zhao, Zhe Chen, Donald Metzler, and Jingzheng Qin. Multitask mixture of sequential experts for user activity streams. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pages 3083–3091, 2020. 3


