Model LEGO: Creating Models Like Disassembling and Assembling Building Blocks

Jiacong Hu, Jing Gao, Jingwen Ye, Yang Gao, Xingen Wang, Zunlei Feng*, Mingli Song

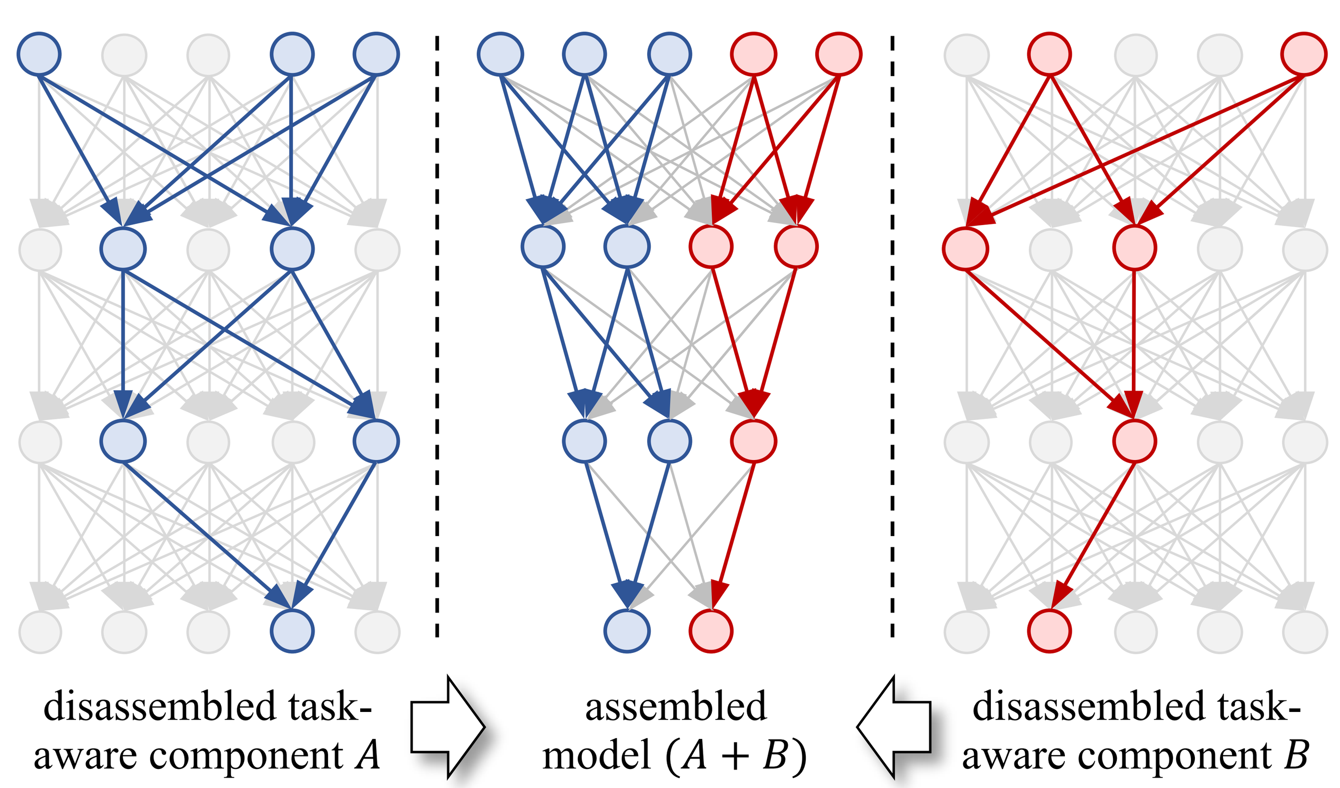
ZHEJIANG UNIVERSITY

NEURAL INFORMATION PROCESSING SYSTEMS

1. Introduction

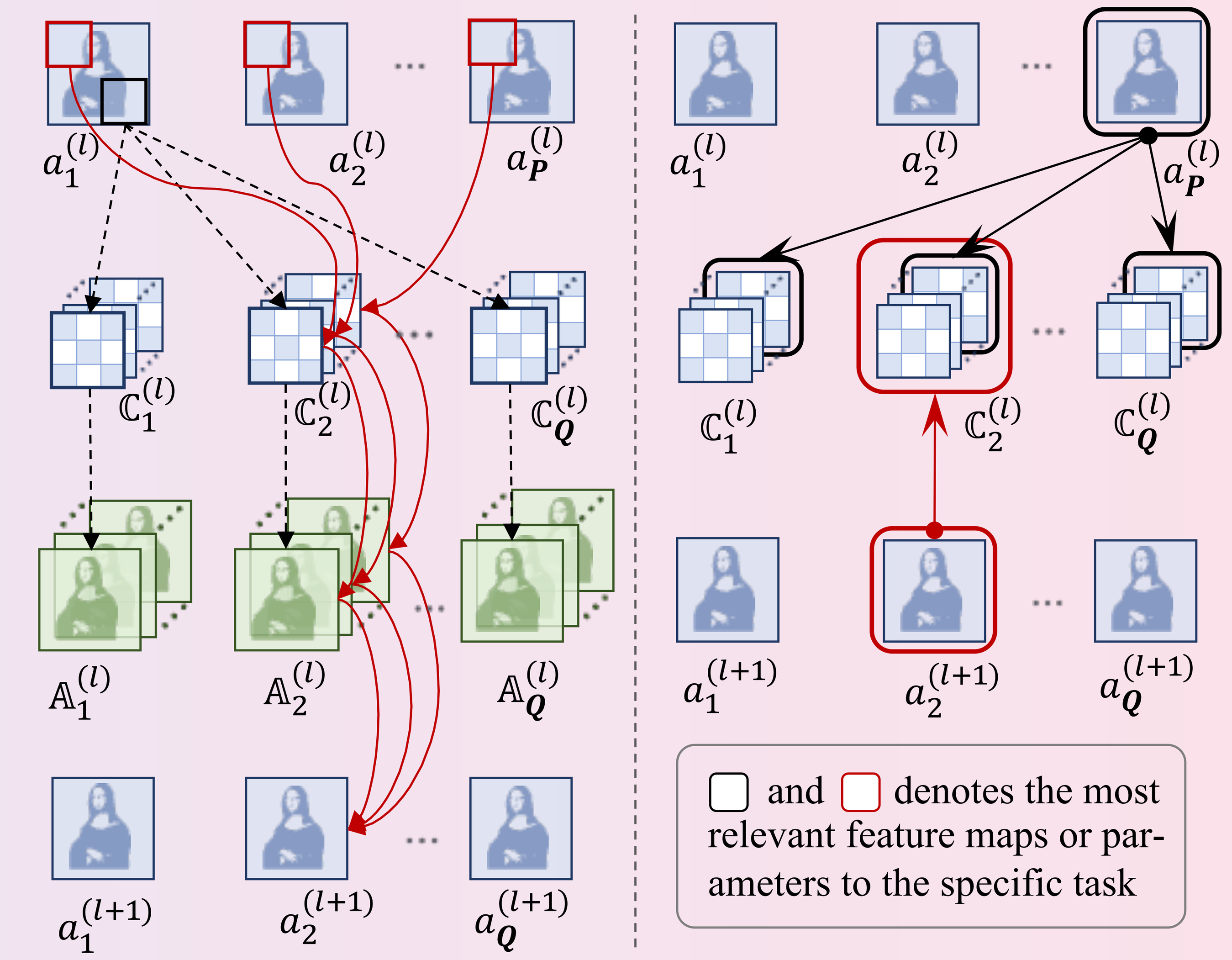
As deep learning models become increasingly complex, training new models from scratch has grown significantly more resource-intensive. Drawing inspiration from the information processing pathways of the biological visual system, we propose a novel paradigm called ***Model Disassembling and Assembling*** (**MDA**). This approach enables the construction of new models without additional training, akin to assembling LEGO bricks. Specifically, MDA allows trained models to be deconstructed into task-aware components, which can then be recombined to create models tailored to specific tasks.

2. Model Disassembling and Assembling



- Let $\mathcal{M}^{(n)}$ denote a trained model with subtasks $\{t_k^{(n)}\}_{k=1}^{K^{(n)}}$.
- The disassembling process extracts task-specific components $\mathcal{M}\left[t_k^{(n)}\right]$ from a trained model $\mathcal{M}^{(n)}$ that contains parameters uniquely relevant to subtask $t_k^{(n)}$.
- The assembling process integrates task-specific components from multiple models to form a new model $\mathcal{M}^{(new)} = \mathcal{M}\left[t_{\{k_1\}}^{\{(n_1)\}}, \ldots, t_{\{k_2\}}^{\{(n_1)\}}, t_{\{k_3\}}^{\{(n_2)\}}, \ldots, t_{\{k_4\}}^{\{(n_2)\}}\right]$, preserving functionality for subtasks $\{t_{k_1}^{(n_1)}, \ldots, t_{k_2}^{(n_1)}\}$ and $\{t_{k_3}^{(n_2)}, \ldots, t_{k_4}^{(n_2)}\}$.

3. Model Disassembling



(a) Contribution Aggregation and Allocation

(b) Component Locating Technique

Contribution Aggregation and Allocation

Taking a CNN classifier as an example, we treat each category as a subtask, with relative contributions quantifying the influence of individual feature maps on the predicted outcomes. This concept extends across all layers to assess contributions from each feature map throughout the network. For a layer l with P input channels and Q output channels, each filter $\mathbb{C}_q^{(l)} = \{c_{q,p}^{(l)}\}_{p=1}^P$ generates corresponding hidden feature maps $\mathbb{A}_q^{(l)}$ and output feature maps $a_q^{(l+1)}$:

$$a_q^{(l+1)} = \sum_{p=1}^{P} a_{q,p}^{(l)} + b_q^{(l)}, \mathbb{A}_q^{(l)} = \{a_{q,p}^{(l)}\}_{p=1}^P, a_{q,p}^{(l)} = c_{q,p}^{(l)} \otimes a_p^{(l)}$$

Contribution Aggregation. The normalized relative contribution $r_{q,p}^{(l)}$ is derived from the positive contributions of the hidden feature map $a_{q,p}^{(l)}$:

$$r_{q,p}^{(l)} = \frac{\hat{s}_{q,p}^{(l)} - \min\left(\hat{s}_{q,p}^{(l)}\right)}{\max\left(\{\hat{s}_{q,p}^{(l)}\}_{p=1}^P\right) - \min\left(\{\hat{s}_{q,p}^{(l)}\}_{p=1}^P\right) + \epsilon}, \hat{s}_{q,p}^{(l)} = \max\left(s_{q,p}^{(l)}, 0\right), s_{q,p}^{(l)} = \sum_{h=1}^{H^{(l)}} \sum_{w=1}^{W^{(l)}} a_{q,p}^{(l)}[h,w]$$

Contribution Allocation. The relative contribution $r_p^{(l)}$ is computed based on the positive contributions from $a_p^{(l)}$:

$$r_p^{(l)} = \frac{\hat{s}_p^{(l)} - \min\left(\{\hat{s}_p^{(l)}\}\right)}{\max\left(\{\hat{s}_p^{(l)}\}\right) - \min\left(\{\hat{s}_p^{(l)}\}\right) + \epsilon}, \hat{s}_p^{(l)} = \max\left(s_p^{(l)}, 0\right), s_p^{(l)} = \sum_{q=1}^{Q} r_{q,p}^{(l)}$$

Component Locating Technique

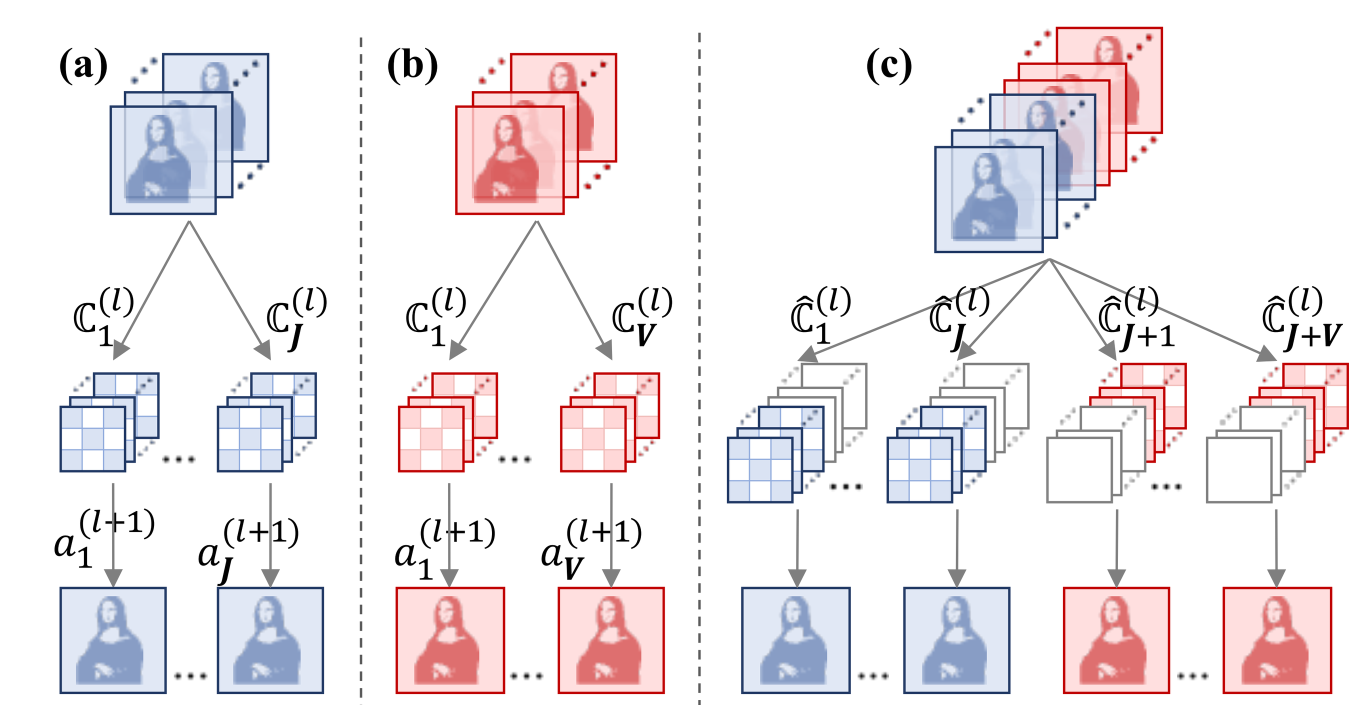
Relevant Feature Identification. The relative contributions $r_{q,p}^{(l)}$ and $r_p^{(l)}$ are transformed into binary hard relative contributions $\hat{r}_{q,p}^{(l)}$ and $\hat{r}_p^{(l)}$ indicating whether the corresponding feature maps are related to the subtask.

$$\hat{r}_{q,p}^{(l)} = \begin{cases} 1, & r_{q,p}^{(l)} \geq \alpha \\ 0, & r_{q,p}^{(l)} < \alpha \end{cases}, \hat{r}_p^{(l)} = \begin{cases} 1, & r_p^{(l)} \geq \beta \\ 0, & r_p^{(l)} < \beta \end{cases}$$

Parameter Linking. If the feature map $a_p^{(l)}$ is identified as most relevant to the predicted result, then the convolution kernels $\{c_{q,p}^{(l)}\}_{p=1}^P$ associated with it are also deemed most relevant to the predicted result.

$$a_q^{(l+1)} = \sum_{p=1}^{P} c_{q,p}^{(l)} \otimes a_p^{(l)} + b_q^{(l)}$$

4. Model Assembling



Alignment Padding Strategy

This strategy introduces empty kernels to ensure that all filters from different disassembled models are aligned. Specifically, each filter in each disassembled model contains $I + U$ kernels:

$$\hat{\mathbb{C}}_j^{(l)} = \{c_{j,1}^{(l)}, \ldots, c_{j,I}^{(l)}, 0, \ldots, 0_U\} \text{ and } \hat{\mathbb{C}}_v^{(l)} = \{0, \ldots, 0_I, c_{v,1}^{(l)}, \ldots, c_{v,U}^{(l)}\}$$

Parameter Scaling Strategy

Scales filters to balance feature magnitudes, thereby preventing bias across the combined components. For a layer l, we calculate the individual and average magnitudes of each feature map as follows:

$$\bar{e}^{(l+1)} = \frac{1}{J+V}\left(\sum_{j=1}^{J} e_j^{(l+1)} + \sum_{v=1}^{V} e_v^{(l+1)}\right), e_j^{(l+1)} = \sum_{h=1}^{H^{(l+1)}} \sum_{w=1}^{W^{(l+1)}} a_j^{(l+1)}[h,w]$$

Then, each filter $\mathbb{C}_j^{(l)}$ is scaled to match the average magnitude: $\tilde{\mathbb{C}}_j^{(l)} = \left(\frac{\bar{e}^{(l+1)}}{e_j^{(l+1)}}\right)\mathbb{C}_j^{(l)}$.

5. Experiments

The Result of Model Disassembling

Dataset	Disassembled Task	VGG-16 Base. (%)	VGG-16 Disa. (%)	ResNet-50 Base. (%)	ResNet-50 Disa. (%)	GoogleNet Base. (%)	GoogleNet Disa. (%)
CIFAR-10	0	94.40	100.00 (*+5.60*)	96.10	100.00 (*+3.90*)	95.40	100.00 (*+4.60*)
	1	96.50	100.00 (*+3.50*)	96.20	100.00 (*+3.80*)	97.40	100.00 (*+2.60*)
	0-2	93.87	95.47 (*+1.60*)	94.93	97.43 (*+2.50*)	94.47	98.17 (*+3.70*)
	3-9	92.49	92.27 (*-0.22*)	93.81	94.46 (*+0.65*)	93.46	93.17 (*-0.29*)
CIFAR-100	0	84.00	100.00 (*+16.00*)	92.00	100.00 (*+8.00*)	90.00	100.00 (*+10.00*)
	1	87.00	100.00 (*+13.00*)	87.00	100.00 (*+13.00*)	85.00	100.00 (*+15.00*)
	0-19	71.05	82.50 (*+11.45*)	75.55	77.15 (*+1.60*)	75.90	87.55 (*+11.65*)
	20-69	72.74	79.66 (*+6.92*)	77.68	79.72 (*+2.04*)	76.60	82.42 (*+5.82*)
Tiny-ImageNet	0	82.00	100.0 (*+18.00*)	92.00	100.00 (*+8.00*)	88.00	100.00 (*+12.00*)
	1	70.00	100.0 (*+30.00*)	80.00	100.00 (*+20.00*)	76.00	100.00 (*+24.00*)
	0-69	50.17	55.49 (*+5.32*)	56.06	56.40 (*+0.34*)	52.89	59.40 (*+6.51*)
	70-179	45.36	47.95 (*+2.59*)	51.27	53.42 (*+2.15*)	47.91	51.04 (*+3.13*)

The Result of Model Assembling

Dataset	Assembled Task	VGG-16 Base. (%)	VGG-16 Asse. (%)	ResNet-50 Base. (%)	ResNet-50 Asse. (%)	GoogleNet Base. (%)	GoogleNet Asse. (%)
CIFAR-10 + CIFAR-100	0 + 0	89.20	87.00 / 88.43	94.05	77.30 / 87.36	99.60	94.25 / 95.27
	0-2 + 0-19	74.03	74.17 / 74.19	78.08	64.34 / 76.37	78.32	79.22 / 79.22
	3-9 + 20-69	75.16	73.72 / 74.25	79.66	72.03 / 75.25	78.67	70.24 / 76.37
	0-9 + 20-99	74.87	72.07 / 73.18	79.54	65.97 / 74.65	78.57	66.59 / 70.36
CIFAR-10 + Tiny-ImageNet	0 + 0	88.20	94.70 / 90.72	94.05	86.95 / 94.51	91.70	62.40 / 80.34
	0-2 + 0-69	51.97	53.20 / 53.20	57.65	43.09 / 56.38	54.59	57.51 / 57.81
	3-9 + 0-69	54.02	50.20 / 52.48	59.49	52.14 / 58.63	56.57	53.74 / 55.32
	0-9 + 70-179	49.33	42.30 / 47.98	54.85	47.21 / 55.17	51.73	48.00 / 52.16
CIFAR-100 + Tiny-ImageNet	0 + 0	83.00	50.00 / 76.28	92.00	53.00 / 87.34	89.00	50.00 / 85.28
	0-19 + 0-69	69.86	50.66 / 57.19	74.59	57.86 / 69.23	74.73	58.67 / 69.14
	20-69 + 70-179	71.48	50.08 / 65.71	76.54	56.53 / 69.79	75.08	54.09 / 71.27
	0-99 + 0-199	55.97	43.06 / 56.13	61.51	53.05 / 57.23	59.19	48.66 / 58.28