1. Dataset Details

MiniImageNet [10] is designed to build a lightweight but challenging dataset. It consists of 600 RGB images per class from 100 different classes from ILSVRC-12 [9]. And the classes are randomly sampled from ImageNet, which has 1000 classes. As in previous work [2-4, 6], we adopt the same split of Ravi & Larochelle [7], who resized all images to $84 \times 84$ and use 64 classes for training, 16 for validation and 20 for testing.

TieredImageNet [8] is also a subset of ILSVRC-12. Different from miniImageNet, tieredImageNet is not randomly sampled. It extracts 34 categories from ILSVRC-12, each of which contains $10 \sim 30$ different classes, with a total of 608 classes. Each classes has a varying number of images, with a total of 779,165 images. Unlike miniImageNet, tieredImageNet considers ImageNet’s category hierarchy. The data is divided according to categories, in which 20 categories (351 classes, 448,695 images) are used as training sets, 6 categories (97 classes, 124,261 images) are used as validation sets, and 8 categories (160 classes, 206,209 images) are used as test sets. We use the same split as [8] and resized all images to $84 \times 84$ as the same as previous work [2-4].

CIFAR-FS [1] (CIFAR100 few shots) is randomly sampled from CIFAR100 [5] by using the same criteria with miniImageNet. We use the same split as [1]. But it is much more lightweight. It also consists of 600 RGB images per class from 100 different classes and use 64 classes for training, 16 for validation and 20 for testing. But the resolution of the image is only $32 \times 32$.

2. Method Details

In order to illustrate our proposed method more clearly, as shown in Algorithms 1 and 2, we give the PyTorch like pseudo code of our proposed pixel-wise similarity module and energy-based module, respectively.

```
# f_query: Tensor [2NQ, 1, m^2, dim/2, 1]
# is the feature map of all query samples
# f_proto: Tensor [1, N, 1, dim/2, m^2]
# is the feature map of all classes
# N: N-way
# Q: Q-query
# m: spatial dimension of feature map
# dim: channel dimension of feature map
# top_k: the hyperparameter used to select topk
# temp: temperature coefficient hyperparameter
class PixelSimilarity(Module):
    def __init__(self, args):
        self.top_k = args.top_k
        self.temperature = args.temp
        self.cos = nn.CosineSimilarity(dim=3)
    
    def forward(f_query, f_proto):
        # cosine similarity
        sim = self.cos(f_query, f_proto)
        # topk
        sim = sim.topk(self.top_k, dim=3).values
        sim = sim / self.temperature
        return sim
```

3. Experiments Details

3.1. Detailed Task Sampling

For a N-way K-shot Q-query FSOR task, its sampling is independent of the dataset and training or testing stage. Specifically, it will randomly select 2N classes from the dataset, half of which will be regarded as known classes and the other half as unknown classes. Q samples will be sampled for each classes as query, while additional K samples will be sampled for known classes as support.
Algorithm 2: Energy-based Module, PyTorch like

```python
# e_sim: Tensor [2NQ, N] is class-wise similarity
# f_sim: Tensor [2NQ, N] is pixel-wise similarity
# N: N-way
# Q: Q-query
# m_k: margin for closed-set samples
# m_u: margin for open-set samples
class EnergyLoss(Module):
    def __init__(self, args):
        self.m_k = args.m_k
        self.m_u = args.m_u

    def forward(e_sim, f_sim):
        # energy score
        e_ergy = -torch.logsumexp(e_sim, dim=1)
        f_ergy = -torch.logsumexp(f_sim, dim=1)
        energy = e_ergy + f_ergy

        # energy loss
        k_ergy, u_ergy = torch.split(energy, NQ)
        l_k = pow(F.relu(k_ergy - self.m_k), 2)
        l_u = pow(F.relu(self.m_u - u_ergy), 2)
        l_energy = l_k.mean() + l_u.mean()
        return l_energy
```

3.2. Detailed Ablation Study

We provide the detailed ablation results on miniImageNet,tieredImageNet and CIFAR-FS under 5-way 1-shot and 5-shot setting.

As show in Table 1, we give the ablation study results of three modules proposed by us on all benchmarks. On miniImageNet and tieredImageNet dataset, our three modules can improve the performance of open-set recognition while maintaining the closed-set classification performance. On the CIFAR-FS dataset, our pixel-wise similarity branch may not work due to the decrease in original image resolution. The spatial dimension of the feature map is only $2 \times 2$, the pixel-wise similarity branch is difficult to get discriminative local information. But using only the energy loss, our open-set recognition performance can significantly exceed the baseline.

We also give the results of glocal energy score combination of the class-wise branch and pixel-wise on all benchmarks. It can be seen in Table 2 that Ahead Combine can always improve open-set recognition performance compared to Delay Combine.

For the ablation study of combination coefficients between class-wise energy and pixel-wise energy, we give detailed results in Table 3. Compared to Learnable and Task-adaptive, Fixed value achieves better open-set recognition performance on all benchmarks.

References

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>5-way 1-shot ACC</th>
<th>5-way 1-shot AUROC</th>
<th>5-way 5-shot ACC</th>
<th>5-way 5-shot AUROC</th>
</tr>
</thead>
<tbody>
<tr>
<td>miniImageNet</td>
<td>Delay Combine</td>
<td>68.39 ± 0.86</td>
<td>73.17 ± 0.83</td>
<td>83.00 ± 0.55</td>
<td>80.67 ± 0.65</td>
</tr>
<tr>
<td></td>
<td>Ahead Combine</td>
<td>68.26 ± 0.85</td>
<td>73.70 ± 0.82</td>
<td>83.05 ± 0.55</td>
<td>82.29 ± 0.60</td>
</tr>
<tr>
<td>tieredImageNet</td>
<td>Delay Combine</td>
<td>70.72 ± 0.99</td>
<td>74.00 ± 0.79</td>
<td>84.44 ± 0.66</td>
<td>80.57 ± 0.69</td>
</tr>
<tr>
<td></td>
<td>Ahead Combine</td>
<td>70.50 ± 0.93</td>
<td>75.86 ± 0.81</td>
<td>84.60 ± 0.65</td>
<td>81.95 ± 0.72</td>
</tr>
<tr>
<td>CIFAR-FS</td>
<td>Delay Combine</td>
<td>76.58 ± 0.86</td>
<td>78.29 ± 0.76</td>
<td>87.75 ± 0.64</td>
<td>86.01 ± 0.63</td>
</tr>
<tr>
<td></td>
<td>Ahead Combine</td>
<td>76.77 ± 0.88</td>
<td>78.67 ± 0.80</td>
<td>86.74 ± 0.66</td>
<td>86.56 ± 0.59</td>
</tr>
</tbody>
</table>

Table 2. Ablation study of global energy score combination of the class-wise branch and pixel-wise branch on miniImageNet, tieredImageNet and CIFAR-FS under 5-way 1-shot and 5-shot setting.
### Table 3. Ablation study of combination coefficients between class-wise energy and pixel-wise energy on miniImageNet, tieredImageNet and CIFAR-FS under 5-way 1-shot and 5-shot setting.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>5-way 1-shot</th>
<th>5-way 5-shot</th>
<th>5-way 1-shot</th>
<th>5-way 5-shot</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>ACC</td>
<td>AUROC</td>
<td>ACC</td>
<td>AUROC</td>
</tr>
<tr>
<td>miniImageNet</td>
<td>Fixed value</td>
<td>68.26 ± 0.85</td>
<td>73.70 ± 0.82</td>
<td>83.05 ± 0.55</td>
<td>82.29 ± 0.60</td>
</tr>
<tr>
<td></td>
<td>Learnable</td>
<td>68.34 ± 0.85</td>
<td>73.40 ± 0.82</td>
<td>83.08 ± 0.56</td>
<td>81.37 ± 0.64</td>
</tr>
<tr>
<td></td>
<td>Task-adaptive</td>
<td>68.34 ± 0.85</td>
<td>73.71 ± 0.72</td>
<td>83.00 ± 0.56</td>
<td>81.34 ± 0.62</td>
</tr>
<tr>
<td>tieredImageNet</td>
<td>Fixed value</td>
<td>70.50 ± 0.93</td>
<td>75.86 ± 0.81</td>
<td>84.60 ± 0.65</td>
<td>81.95 ± 0.72</td>
</tr>
<tr>
<td></td>
<td>Learnable</td>
<td>70.99 ± 0.93</td>
<td>75.59 ± 0.80</td>
<td>84.84 ± 0.65</td>
<td>80.86 ± 0.70</td>
</tr>
<tr>
<td></td>
<td>Task-adaptive</td>
<td>70.68 ± 0.92</td>
<td>74.93 ± 0.83</td>
<td>84.79 ± 0.66</td>
<td>80.49 ± 0.72</td>
</tr>
<tr>
<td>CIFAR-FS</td>
<td>Fixed value</td>
<td>76.77 ± 0.88</td>
<td>78.67 ± 0.80</td>
<td>86.74 ± 0.66</td>
<td>86.56 ± 0.59</td>
</tr>
<tr>
<td></td>
<td>Learnable</td>
<td>76.60 ± 0.87</td>
<td>78.62 ± 0.80</td>
<td>86.94 ± 0.65</td>
<td>87.29 ± 0.57</td>
</tr>
<tr>
<td></td>
<td>Task-adaptive</td>
<td>76.70 ± 0.87</td>
<td>77.99 ± 0.78</td>
<td>87.33 ± 0.60</td>
<td>85.67 ± 0.60</td>
</tr>
</tbody>
</table>

### Table 4. Results of our method with ResNet-18 backbone. We reimplement our method using ResNet-18 as RFDNet, and report their comparison in the table. Our method still outperforms RFDNet in most cases.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>5-way 1-shot</th>
<th>5-way 5-shot</th>
<th>5-way 1-shot</th>
<th>5-way 5-shot</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>ACC</td>
<td>AUROC</td>
<td>ACC</td>
<td>AUROC</td>
</tr>
<tr>
<td>MiniImageNet</td>
<td>RFDNet</td>
<td>66.40 ± 0.82</td>
<td>71.91 ± 0.78</td>
<td>81.91 ± 0.60</td>
<td>81.29 ± 0.56</td>
</tr>
<tr>
<td></td>
<td>Ours</td>
<td>67.40 ± 0.84</td>
<td>72.35 ± 0.80</td>
<td>82.32 ± 0.56</td>
<td>80.89 ± 0.65</td>
</tr>
<tr>
<td>TieredImageNet</td>
<td>RFDNet</td>
<td>70.33 ± 0.90</td>
<td>75.12 ± 0.74</td>
<td>84.88 ± 0.62</td>
<td>82.19 ± 0.66</td>
</tr>
<tr>
<td></td>
<td>Ours</td>
<td>70.45 ± 0.92</td>
<td>75.69 ± 0.83</td>
<td>84.21 ± 0.67</td>
<td>81.47 ± 0.76</td>
</tr>
<tr>
<td>CIFAR-FS</td>
<td>RFDNet</td>
<td>71.34 ± 0.90</td>
<td>76.74 ± 0.75</td>
<td>84.61 ± 0.62</td>
<td>85.17 ± 0.57</td>
</tr>
<tr>
<td></td>
<td>Ours</td>
<td>73.36 ± 0.89</td>
<td>77.69 ± 0.82</td>
<td>85.46 ± 0.63</td>
<td>85.68 ± 0.64</td>
</tr>
</tbody>
</table>
Figure 1. Four additional examples.