A Dataset Details

Miniimagenet [56] A dataset selected from ImageNet with 100 different classes, each with 600 images. All images are the same size of 84×84. We adopt the same splits of [46] with meta train/validation/test splits being 64/16/20 classes.

CIFARFS [12] A dataset randomly sampled from CIFAR-100. The meta train/validation/test splits are of 64/16/20 classes respectively. We follow the split of [12].

Omniglot [32] An image dataset of 1623 handwritten characters from 50 different alphabets, with 20 examples per class. We follow the setup and data split in [56] and use this dataset as OOD data.

CUB [67] A dataset for fine-grained classification on 200 different bird species. The meta train/validation/test splits are of 100/50/50 classes respectively. We follow the same split of [17].

AIRCRAFT [38] A dataset of images for aircrafts model classification consisting of 102 categories, with 100 images per class. The dataset is split into 70/15/15 classes for meta- training/validation/test. We follow the split in [59].

Plantae [28] A large dataset consisting of approximately 100K images with 200 randomly selected plant species. We follow the split of [55] and split the dataset into 100/50/50 classes for meta- training/validation/test.

Butterfly [16] A large scale fine-grained dataset of Butterfly, they have collected 25279 butterfly images from 200 species, with each species containing at least 30 images. We only keep the classes with more than 50 images and got 185 classes. We randomly split the dataset into 100/40/45 classes for meta-training/validation/test.

Vggflower [43] there are 102 flower categories and contains between 40 and 258 images for each class. We use this dataset for OOD data.

Fish [70] A large scale dataset of fish species. It consists of 1000 fish categories with a total of 54,459 images, we use this dataset for OOD data.

B Implementation details

Memory buffer update

The memory is updated by online reservoir sampling (RS) [58], at each time step t, RS operates on task unit level. RS ensures each task has the same probability to be stored in the memory buffer without knowing the total number of tasks in advance. The algorithm works by maintaining a reservoir of size k tasks. As long as the memory buffer is not full, each new incoming task will be

stored in the memory buffer. When the memory buffer is full, at time step i, when the task i arrives, the algorithm then generates a random number j between (and including) 1 and i. If j is at most k, then the task is selected and replaces one task in the reservoir. Otherwise, the task i is discarded. The memory buffer is updated online and real time. Whether each task is determined to be stored in the memory or not depends on whether the random number j is less than k or not.

Implementation details. We use a five-layer CNN with 64 filters of kernel size being 3 for meta-learning. Similar architecture is commonly used in existing meta-learning literature. We do not use any pre-trained network feature extractors, which assume all the training tasks data are available before training, and this violates our problem setting that future incoming tasks are completely unknown. Methods are evaluated on the proposed benchmark described in Section 3. We perform experiments on different dataset ordering, with the default ordering being Plantae, CUB, MiniImagenet, CIFARFS, Aircraft, Butterfly. For each dataset, we randomly sample 12K tasks, we thus have 72K tasks in total. This task sequence is much larger than existing continual learning models considered. The evolving episodes are constructed and described in Section 3. All experiments are averaged over five independent runs. We create an additional split for each dataset to separate the images of each class into disjoint labeled and unlabeled sets. Following [47], we sample 40% of the images of each class to form the labeled split, and the remaining 60% can only be used in the unlabeled portion of episodes. We randomly sample 10 data points from the unlabeled portion of each chosen class to form the unlabeled ID data $\mathcal{U}_{id}$ for each episode. The unlabeled OOD data $\mathcal{U}_{ood}$ for each episode is produced by sampling R unlabeled data from the mixture OOD dataset of *Omniglot* [32], *Vggflower* [43] and *Fish* [70]. $R = 50$ by default. This way of constructing multimodal OOD data is more challenging and more realistic than existing SSFSL works, which sample OOD data from the same meta training dataset. More implementation details are given in Appendix B. We provide code in supplementary materials.

We randomly sample 12000 tasks from each dataset. We sequentially train on each dataset for 6000 iterations in the dataset sequence. At each iteration, we randomly sample 2 tasks (meta batch size) from current dataset. The hyper-parameter is determined by grid search, with $\lambda = \{1e-4, 1e-5, 1e-6\}$ and $\beta = \{0.01, 0.005, 0.003, 0.001\}$. The grid search results for 5-way 5-shot learning are in Table 8. The adopted values are $\lambda = 1e-5$ and $\beta = 0.5$. The OOD sampling is performed by uniformly sampling equal number of OOD data from each OOD dataset. We use Adam optimizer [30] to optimize the model with learning rate of 1e-3. For theorem 3, we set $a(y) = 1$ for simplicity. The function in $f(\mathbf{e}_{id}, \mathbf{p}_{c'})$ is one layer network with 10 hidden units.

C More experimental results

Effect of domain ordering

Order 2 : Butterfly, CUB, CIFARFS, Plantae, MiniImagenet, Aircraft. 5-way, 1-shot and 5-shot learning results are shown in Table 4 and Table 5.

Order 3 : CUB, CIFARFS, Plantae, MiniImagenet, Butterfly, Aircraft. Results are shown in Table 6 and Table 7

Table 4: 5-way, 1-shot and 5-shot classification accuracy comparing to meta-learning baselines with domain order 2. Top rows are meta learning *without unlabeled data*, bottom rows are meta learning methods *with unlabeled data*.

Algorithm	1-Shot	5-Shot
ProtoNet	27.91 ± 0.95	39.12 ± 0.76
ANIL	27.25 ± 0.82	38.58 ± 0.65
MSKM	30.15 ± 0.89	41.02 ± 0.75
LST	32.51 ± 0.71	41.78 ± 0.85
TPN	31.81 ± 0.72	42.81 ± 0.62
Ours	$\mathbf{42.49 \pm 0.57}$	$\mathbf{55.37 \pm 0.71}$

Table 5: 5-way, 1-shot and 5-shot classification accuracy compared to continual learning baselines.

Algorithm	1-Shot	5-Shot
Semi-ER	39.05 ± 0.91	51.28 ± 0.78
Semi-AGEM	39.48 ± 0.97	51.51 ± 0.83
Semi-MER	39.65 ± 0.82	51.42 ± 0.68
Semi-GPM	38.89 ± 0.85	51.03 ± 0.71
Semi-DEGCL	39.81 ± 0.73	51.68 ± 0.62
Ours	$\mathbf{42.49 \pm 0.57}$	$\mathbf{55.37 \pm 0.71}$
Joint-training	49.91 ± 0.79	61.78 ± 0.75

Table 6: 5-way, 1-shot and 5-shot classification accuracy comparing to meta-learning baselines with domain order 2. Top rows are meta learning *without unlabeled data*, bottom rows are meta learning methods *with unlabeled data*.

Algorithm	1-Shot	5-Shot
ProtoNet	28.55 ± 1.03	40.41 ± 0.97
ANIL	28.24 ± 1.15	39.97 ± 0.83
MSKM	31.17 ± 0.96	41.46 ± 0.69
LST	31.61 ± 0.81	41.31 ± 0.80
TPN	32.80 ± 0.71	42.83 ± 0.71
Ours	$\mathbf{43.36 \pm 0.61}$	$\mathbf{56.89 \pm 0.58}$

Table 7: 5-way, 1-shot and 5-shot classification accuracy compared to continual learning baselines.

Algorithm	1-Shot	5-Shot
Semi-ER	38.87 ± 0.76	52.75 ± 0.82
Semi-AGEM	39.29 ± 0.73	52.97 ± 0.94
Semi-MER	39.41 ± 0.68	53.08 ± 0.87
Semi-GPM	39.09 ± 0.81	52.83 ± 0.76
Semi-DEGCL	39.46 ± 0.77	53.37 ± 0.73
Ours	$\mathbf{43.36 \pm 0.61}$	$\mathbf{56.89 \pm 0.58}$
Joint-training	49.91 ± 0.79	61.78 ± 0.75

Table 8: Sensitivity analysis on hyper parameters

λ \ β	0.001	0.003	0.005	0.01
1e-4	52.86 ± 0.79	53.28 ± 0.75	53.49 ± 0.67	52.65 ± 0.79
1e-5	52.61 ± 0.85	53.96 ± 0.71	53.60 ± 0.82	52.73 ± 0.93
1e-6	51.85 ± 0.58	52.06 ± 0.49	52.15 ± 0.70	52.03 ± 0.67

Sensitivity analysis on hyper parameters

We performed sensitivity analysis on hyperparameters, the result is summarized in Table 8. β controls the magnitude of optimal transport regularization, the result indicate the model performance is positively correlated with β when it

increases until optimal trade-off is reached, after which performance deteriorates with over regularization when β reaches 0.01. Similar trend is observed in mutual information regularization with λ.

Effect of memory size

Table 9 shows the effect of different memory size. With memory size increases, the model performance also increases accordingly.

Table 9: 5-way, 1-shot and 5-shot classification accuracy with different memory size.

Memory size	1-Shot	5-Shot
50	40.11 ± 0.73	52.75 ± 0.77
100	40.79 ± 0.58	53.28 ± 0.65
200	41.25 ± 0.64	53.96 ± 0.71

Ablation Study results

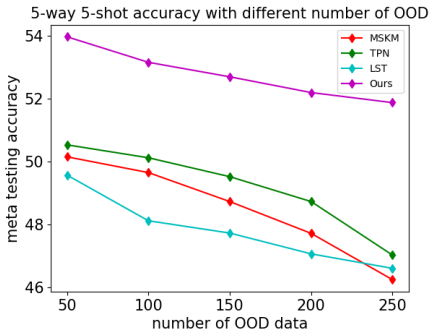

Fig. 2: Sensitivity of baselines and our method to OOD data

Table 10: Running time (seconds).

Algorithm	5-Shot (250 OOD)	5-Shot (50 OOD)
MSKM	**85**	**43**
TPN	149	86
LST	137	79
ORDER (Ours)	122	71

Efficiency Evaluation To investigate the computational efficiency, we report the time cost of baselines and our method running on 200 iterations. The results are reported in Table 10. Our method is more efficient than TPN and LST, but slower than MSKM.

Table 11: Ablation study of model components.

MI	OT	ACC (5 shot 50 OOD)	ACC (5 shot 250 OOD)
✗	✗	50.35 ± 0.76	46.25 ± 0.91
✗	✓	51.18 ± 0.82	47.69 ± 0.87
✗	✓($\mathcal{U}_{id}$)	51.72 ± 0.61	48.57 ± 0.80
✓	✗	52.63 ± 0.77	50.23 ± 0.62
✓	✓($\mathcal{U}_{id}$)	53.96 ± 0.71	51.87 ± 0.68

Table 12: Fine-grained ablation study analysis of using different parts of unlabeled data and mutual information regularizer.

$\mathcal{U}_{id}$	$\mathcal{U}_{ood}$	$\mathcal{I}_{id}$	$\mathcal{I}_{ood}$	ACC (5 shot 50 OOD)	ACC (5 shot 250 OOD)
✓	✓	✗	✗	50.35 ± 0.76	46.25 ± 0.91
✓	✗	✗	✗	51.67 ± 0.85	48.36 ± 0.82
✓	✗	✓	✗	52.87 ± 0.89	49.98 ± 0.70
✓	✓	✓	✓	53.96 ± 0.71	51.87 ± 0.68

D Theorem Proof

Theorem 1 For a task $\mathcal{T} = \{\mathcal{S}, \mathcal{U}, \mathcal{Q}\}$, suppose the unlabeled OOD data embedding $\mathbf{e}_{ood} = h_{\boldsymbol{\theta}}(\mathbf{x}), \mathbf{x} \in \mathcal{U}_{ood}$. The nearest prototype corresponds to $\mathbf{e}_{ood}$ is $\mathbf{p}_{c'}$, where $c' = \text{argmin}_c \,||\, \mathbf{e}_{ood} - \mathbf{p}_c \,||$. Given a collection of samples $\{(\mathbf{e}^i_{ood}, \mathbf{p}^i_{c'})\}_{i=1}^{i=L} \sim P(\mathbf{e}_{ood}, \mathbf{p}_{c'})$, the variational upper bound of mutual information $\mathcal{I}(\mathbf{e}_{ood}, \mathbf{p}_{c'})$ is

$$
\begin{aligned}
&\mathcal{I}(\mathbf{e}_{ood}, \mathbf{p}_{c'}) \\
&\leq \sum_{i=1}^{i=L} \log P(\mathbf{p}^i_{c'} \,|\, \mathbf{e}^i_{ood}) - \sum_{i=1}^{i=L} \sum_{j=1}^{j=L} \log P(\mathbf{p}^i_{c'} \,|\, \mathbf{e}^j_{ood}) \\
&= \mathcal{I}_{ood}. \qquad (10)
\end{aligned}
$$

The bound is tight (equality holds) when $\mathbf{p}_{c'}$ and $\mathbf{e}_{ood}$ are independent.

Proof

$$\begin{aligned}
&\mathbb{E}_{P(\mathbf{e}_{ood},\mathbf{p}_{c'})}[\log P(\mathbf{p}_{c'} \mid \mathbf{e}_{ood})] \\
&\quad - \mathbb{E}_{P(\mathbf{p}_{c'})P(\mathbf{e}_{ood})}[\log P(\mathbf{p}_{c'} \mid \mathbf{e}_{ood})] - \mathcal{I}(\mathbf{e}_{ood}, \mathbf{p}_{c'}) \\
&= \mathbb{E}_{P(\mathbf{e}_{ood},\mathbf{p}_{c'})}[\log P(\mathbf{p}_{c'} \mid \mathbf{e}_{ood})] \\
&\quad - \mathbb{E}_{P(\mathbf{p}_{c'})P(\mathbf{e}_{ood})}[\log P(\mathbf{p}_{c'} \mid \mathbf{e}_{ood})] \\
&\quad - \mathbb{E}_{P(\mathbf{e}_{ood},\mathbf{p}_{c'})}[\log P(\mathbf{p}_{c'} \mid \mathbf{e}_{ood}) - \log P(\mathbf{p}_{c'})] \\
&= \mathbb{E}_{P(\mathbf{e}_{ood},\mathbf{p}_{c'})}[\log P(\mathbf{p}_{c'})] - \mathbb{E}_{P(\mathbf{p}_{c'})P(\mathbf{e}_{ood})}[\log P(\mathbf{p}_{c'} \mid \mathbf{e}_{ood})] \\
&= \mathbb{E}_{P(\mathbf{p}_{c'})}[\log P(\mathbf{p}_{c'}) - \mathbb{E}_{P(\mathbf{e}_{ood})}[\log P(\mathbf{p}_{c'} \mid \mathbf{e}_{ood})]]
\end{aligned}$$

Following [18],

$$P(\mathbf{p}_{c'}) = \mathbb{E}_{P(\mathbf{e}_{ood})}[P(\mathbf{p}_{c'} \mid \mathbf{e}_{ood})] \tag{11}$$

By Jensen's inequality,

$$\log \mathbb{E}_{P(\mathbf{e}_{ood})}[P(\mathbf{p}_{c'} \mid \mathbf{e}_{ood})] \geq \mathbb{E}_{P(\mathbf{e}_{ood})}[\log P(\mathbf{p}_{c'} \mid \mathbf{e}_{ood})]$$

We then get the following bounds:

$$\begin{aligned}
\mathcal{I}(\mathbf{e}_{ood}, \mathbf{p}_{c'}) \leq\ & \mathbb{E}_{P(\mathbf{e}_{ood},\mathbf{p}_{c'})}[\log P(\mathbf{p}_{c'} \mid \mathbf{e}_{ood})] \\
& - \mathbb{E}_{P(\mathbf{p}_{c'})P(\mathbf{e}_{ood})}[\log P(\mathbf{p}_{c'} \mid \mathbf{e}_{ood})]
\end{aligned} \tag{12}$$

Given a collection of samples $\{(\mathbf{e}^i_{ood}, \mathbf{p}^i_{c'})\}_{i=1}^{i=L} \sim P(\mathbf{e}_{ood}, \mathbf{p}_{c'})$, the upper bound becomes:

$$\begin{aligned}
\mathcal{I}(\mathbf{e}_{ood}, \mathbf{p}_{c'}) \leq & \sum_{\mathbf{p}_{c'}, \mathbf{e}_{ood}} \log P(\mathbf{p}_{c'} \mid \mathbf{e}_{ood}) \\
& - \sum_{c=1}^{c=N} \sum_{j=1}^{j=M} \log P(\mathbf{p}_{c'} \mid \mathbf{e}^j_{ood}) = \mathcal{I}_{ood}.
\end{aligned} \tag{13}$$

Lemma 2 For a task $\mathcal{T} = \{\mathcal{S}, \mathcal{U}, \mathcal{Q}\}$, suppose the unlabeled ID data embedding $\mathbf{e}_{id} = h_{\boldsymbol{\theta}}(\mathbf{x}), \mathbf{x} \in \mathcal{U}_{id}$. The nearest prototype corresponds to $\mathbf{e}_{id}$ is $\mathbf{p}_{c'}$, where $c' = \text{argmin}_c \,||\, \mathbf{e}_{id} - \mathbf{p}_c \,||$. Given a collection of samples $\{(\mathbf{e}^i_{id}, \mathbf{p}^i_{c'})\}_{i=1}^{i=L} \sim P(\mathbf{e}_{id}, \mathbf{p}_{c'})$, the variational lower bound of $\mathcal{I}(\mathbf{e}_{id}, \mathbf{p}_{c'})$ is :

$$\mathcal{I}(\mathbf{e}_{id}, \mathbf{p}_{c'}) \geqslant \sum_{i=1}^{i=L} f(\mathbf{e}^i_{id}, \mathbf{p}^i_{c'}) - \sum_{j=1}^{j=L} \log \sum_{i=1}^{i=L} e^{f(\mathbf{e}^i_{id}, \mathbf{p}^j_{c'})}. \tag{14}$$

This bound is tight if $f(\mathbf{e}_{id}, \mathbf{p}_{c'}) = \log P(\mathbf{p}_{c'} \mid \mathbf{e}_{id}) + c(\mathbf{p}_{c'})$.

Proof

With $q(\mathbf{e}_{id} \mid \mathbf{p}_{c'})$ to approximate $P(\mathbf{e}_{id} \mid \mathbf{p}_{c'})$, we can get the following inequality (following the argument in Barber-Agakov lower bound [9]):

$$\begin{aligned}\mathcal{I}(\mathbf{e}_{id}, \mathbf{p}_{c'}) &= \mathbb{E}_{P(\mathbf{e}_{id}, \mathbf{p}_{c'})}[\frac{q(\mathbf{e}_{id} \mid \mathbf{p}_{c'})}{P(\mathbf{e}_{id})}] \\ &\quad + \mathbb{E}_{P(\mathbf{e}_{id})}[\mathbb{KL}(P(\mathbf{e}_{id} \mid \mathbf{p}_{c'}) | q(\mathbf{e}_{id} \mid \mathbf{p}_{c'}))] \\ &\geq \mathbb{E}_{P(\mathbf{e}_{id}, \mathbf{p}_{c'})}[q(\mathbf{e}_{id} \mid \mathbf{p}_{c'}))] + H(\mathbf{e}_{id})\end{aligned}$$

$H(\mathbf{e}_{id})$ is the entropy of the variable $\mathbf{e}_{id}$. If $q(\mathbf{e}_{id} \mid \mathbf{p}_{c'}))$ is specified as:

$$q(\mathbf{e}_{id} \mid \mathbf{p}_{c'}) = \frac{P(\mathbf{e}_{id})}{Z(\mathbf{p}_{c'})} e^{f(\mathbf{e}_{id}, \mathbf{p}_{c'})} \tag{15}$$

where $f(\mathbf{e}_{id}, \mathbf{p}_{c'})$ is a value function parametrized by a neural network. $Z(\mathbf{p}_{c'})$ is a partition function and is specified as following:

$$Z(\mathbf{p}_{c'}) = \mathbb{E}_{P(\mathbf{e}_{id})}[e^{f(\mathbf{e}_{id}, \mathbf{p}_{c'})}] \tag{16}$$

Then, the above lower bound becomes

$$\mathbb{E}_{P(\mathbf{e}_{id}, \mathbf{p}_{c'})}[f(\mathbf{e}_{id}, \mathbf{p}_{c'}))] - \mathbb{E}_{P(\mathbf{p}_{c'})}[\log Z(\mathbf{p}_{c'}))] \tag{17}$$

Given a collection of samples $\{(\mathbf{e}_{id}^i, \mathbf{p}_{c'}^i)\}_{i=1}^{i=L} \sim P(\mathbf{e}_{id}, \mathbf{p}_{c'})$, the bounds become

$$\mathcal{I}(\mathbf{e}_{id}, \mathbf{p}_{c'}) \geqslant \sum_{i=1}^{i=L} f(\mathbf{e}_{id}^i, \mathbf{p}_{c'}^i) - \sum_{j=1}^{j=L} \log \sum_{i=1}^{i=L} e^{f(\mathbf{e}_{id}^i, \mathbf{p}_{c'}^j)}. \tag{18}$$

Based on Lemma 1, we derive a tractable lower bound as Theorem 2.

Theorem 3 For a task $\mathcal{T} = \{\mathcal{S}, \mathcal{U}, \mathcal{Q}\}$, suppose the unlabeled ID data embedding $\mathbf{e}_{id} = h_{\boldsymbol{\theta}}(\mathbf{x}), \mathbf{x} \in \mathcal{U}_{id}$. The nearest prototype corresponds to $\mathbf{e}_{id}$ is $\mathbf{p}_{c'}$, where $c' = \text{argmin}_c \,||\, \mathbf{e}_{id} - \mathbf{p}_c \,||$. Given a collection of samples $\{(\mathbf{e}_{id}^i, \mathbf{p}_{c'}^i)\}_{i=1}^{i=L} \sim P(\mathbf{e}_{id}, \mathbf{p}_{c'})$, the variational lower bound of $\mathcal{I}(\mathbf{e}_{id}, \mathbf{p}_{c'})$ is:

$$\begin{aligned}\mathcal{I}(\mathbf{e}_{id}, \mathbf{p}_{c'}) \geqslant &- \sum_{j=1}^{j=L} \left[\frac{\sum_{i=1}^{i=L} e^{f(\mathbf{e}_{id}^i, \mathbf{p}_{c'}^j)}}{a(\mathbf{p}_{c'}^j)} + \log(a(\mathbf{p}_{c'}^j)) - 1 \right] \\ &+ \sum_{i=1}^{i=L} f(\mathbf{e}_{id}^i, \mathbf{p}_{c'}^i) = \mathcal{I}_{id}.\end{aligned}$$

The bound is tight (equality holds) when $f(\mathbf{e}_{id}, \mathbf{p}_{c'}) = \log P(\mathbf{p}_{c'} \mid \mathbf{e}_{id}) + c(\mathbf{p}_{c'})$ and $a(\mathbf{p}_{c'}) = \mathbb{E}_{p(\mathbf{e}_{id})} e^{f(\mathbf{e}_{id}, \mathbf{p}_{c'})}$. $a(\mathbf{p}_{c'})$ is any function that $a(\mathbf{p}_{c'}) > 0$.

Proof

Define the function $\delta(x, y) = \log(x) - \frac{x}{y} - \log(y) + 1$, where $x > 0, y > 0$

Take the derivative $\frac{\partial \delta(x,y)}{\partial x} = \frac{y-x}{xy}$, when $x < y$, $\frac{\partial \delta(x,y)}{\partial x} > 0$; when $x > y$, $\frac{\partial \delta(x,y)}{\partial x} < 0$. This implies that when $x = y$, $\delta(x, y)$ achieves maximum. Thus, $\delta(x, y) \leq \delta(y, y) = 0$. We can obtain the following inequality [44]

$$\log(x) \leq \frac{x}{y} + \log(y) - 1 \tag{19}$$

By the inequality of 19, we can get the following inequality:

$$\log Z(\mathbf{p}_{c'}) \leq \frac{Z(\mathbf{p}_{c'})}{a(\mathbf{p}_{c'})} + \log(a(\mathbf{p}_{c'})) - 1 \tag{20}$$

Follows from the above lemma, and plug the Equation 20 into Equation 17, this theorem follows

$$\begin{aligned} \mathcal{I}(\mathbf{e}_{id}, \mathbf{p}_{c'}) \geqslant & \mathbb{E}_{p(\mathbf{e}_{id}, \mathbf{p}_{c'})}[f(\mathbf{e}_{id}, \mathbf{p}_{c'})] \\ & - \mathbb{E}_{p(\mathbf{p}_{c'})}[\frac{\mathbb{E}_{p(\mathbf{e}_{id})} e^{f(\mathbf{e}_{id}, \mathbf{p}_{c'})}}{a(\mathbf{p}_{c'})} + \log(a(\mathbf{p}_{c'})) - 1] \\ = & \mathcal{I}_{id}. \end{aligned} \tag{21}$$

Given a collection of samples $\{(\mathbf{e}_{id}^i, \mathbf{p}_{c'}^i)\}_{i=1}^{i=L} \sim P(\mathbf{e}_{id}, \mathbf{p}_{c'})$, the bounds can be obtained in the following:

$$\begin{aligned} \mathcal{I}(\mathbf{e}_{id}, \mathbf{p}_{c'}) \geqslant & -\sum_{j=1}^{j=L} \left[\frac{\sum_{i=1}^{i=L} e^{f(\mathbf{e}_{id}^i, \mathbf{p}_{c'}^j)}}{a(\mathbf{p}_{c'}^j)} + \log(a(\mathbf{p}_{c'}^j)) - 1 \right] \\ & + \sum_{i=1}^{i=L} f(\mathbf{e}_{id}^i, \mathbf{p}_{c'}^i) = \mathcal{I}_{id}. \end{aligned}$$

Then, we obtain the desired bound.