

---

# Supplementary Materials

## Beyond the Seen: Bounded Distribution Estimation for Open-Vocabulary Learning

---

Anonymous Author(s)

Affiliation

Address

email

### 1 Proof of Theorem 1

In the manuscript, we present Theorem 1 to demonstrate that the estimation error of  $p(\mathbf{x}_u, \mathbf{y}_u)$  is upper bounded. Here we provide further derivations of Theorem 1 of the manuscripts. To this end, We first review the defined setting in open-vocabulary learning task, present some lemmas, and their proofs, which are based on PAC-Bayesian Theorems [5, 6].

**Setting.** The data in open environments can be denoted as  $\mathcal{D}_o = \{\mathcal{D}_s, \mathcal{D}_u\}$ , which consists of image-label pairs  $\{(\mathbf{x}_o, \mathbf{y}_o)\} = \{(\mathbf{x}_s, \mathbf{y}_s), (\mathbf{x}_u, \mathbf{y}_u)\}$ . We assume a predicted unseen-class data distribution  $E$ , where  $(\mathbf{x}_e^i, \mathbf{y}_e^i)$  has the probability  $E_i$ . Similarly, the distributions of training data, unseen data and data in open environments are denoted as  $S, U$  and  $O$ , respectively.

**Lemma 1.** *With probability at least  $1 - \delta$  over the training dataset of size  $m$ , we have the following,*

$$\sum S_i e^{(2m-1)\gamma_i^2} \leq \frac{4m}{\delta}. \quad (1)$$

*Proof.* By the Chernoff bound we have  $P(\gamma \geq x) \leq 2e^{-2mx^2}$ . We now consider the density function  $f(\gamma)$  maximizing  $\int_0^\infty e^{(2m-1)\gamma^2} f(\gamma) d\gamma$  subject to the constraint that  $\int_x^\infty f(\gamma) d\gamma \leq 2e^{-2mx^2}$ . The maximum occurs when we have  $\int_x^\infty f(\gamma) d\gamma = 2e^{-2mx^2}$  which is realized when  $f(\gamma) = 8m\gamma e^{-2m\gamma^2}$ . So we have the following.

$$\begin{aligned} \mathbb{E}_S e^{(2m-1)\gamma^2} &\leq \int_0^\infty e^{(2m-1)\gamma^2} f(\gamma) d\gamma \\ &= \int_0^\infty 8m\gamma e^{(2m-1)\gamma^2} e^{-2m\gamma^2} d\gamma \\ &= \int_0^\infty 8m\gamma e^{-\gamma^2} d\gamma \\ &= 4m, \end{aligned} \quad (2)$$

which suffices to the following.

$$\forall i \quad \mathbb{E}_S \left[ e^{(2m-1)\gamma_i^2} \right] \leq 4m. \quad (3)$$

So we have the following.

$$\mathbb{E}_S \mathbb{E}_i e^{(2m-1)\gamma_i^2} \leq 4m. \quad (4)$$

By applying Markov's inequality on Eq. (4), we get the following.

$$P \left[ \sum S_i e^{(2m-1)\gamma_i^2} \geq \frac{4m}{\delta} \right] \leq \delta, \quad (5)$$

18 which suffices to lemma 1.

19 To prove lemma 6 we consider selecting a training data distribution  $\mathbf{S}$  and a predicted unseen-class  
20 data distribution  $\mathbf{E}$ . Lemma 1 implies that with probability at least  $1 - \delta$  we have the following.

$$\sum \mathbf{S}_i e^{(2m-1)\gamma_i^2} \leq \frac{4m}{\delta}. \quad (6)$$

21 So to prove lemma 1 it now suffices to show that Eq. (6) plus  $\ln \frac{1}{\delta} \leq 2m$  implies the following for all  
22 distributions  $\mathbf{E}$  such that  $d(\mathbf{E}, \mathbf{S}) \leq 2m$ .

$$\sum \mathbf{E}_i \gamma_i \leq \sum \mathbf{E}_i \sqrt{\frac{\ln \frac{\mathbf{E}_i}{\mathbf{S}_i} + \ln \frac{1}{\delta} + \frac{5}{2} \ln m + 8}{2m-1}}. \quad (7)$$

23 To prove Eq. (7) for a given  $\mathbf{E}$  we select  $\gamma_i$  so as to maximize the quantity  $\sum \mathbf{E}_i \gamma_i$  subject to the  
24 constraint Eq. (6). Using Lagrange multipliers we set the gradient of the constraint to be equal to a  
25 multiplier  $\lambda$  times the gradient of the objective function.

$$2(2m-1)\gamma_i e^{(2m-1)\gamma_i^2} \mathbf{S}_i = \lambda \mathbf{E}_i. \quad (8)$$

26 Eq. (7) is trivially true if  $\mathbf{E}_i > 0$  but  $\mathbf{S}_i = 0$  for some  $i$ . So we can assume without loss of generality  
27 that  $\mathbf{S}_i > 0$  whenever  $\mathbf{E}_i > 0$ . This allows the above to be rewritten as follows.

$$2(2m-1)\gamma_i e^{(2m-1)\gamma_i^2} = \frac{\lambda \mathbf{E}_i}{\mathbf{S}_i}. \quad (9)$$

28 Note that  $2(2m-1)\gamma e^{(2m-1)\gamma^2}$  is an unbounded monotonically increasing function of  $\gamma$ . We now  
29 define  $\Delta_i(\lambda)$  to be the unique non-negative value satisfying the following.

$$2(2m-1)\Delta_i(\lambda) e^{(2m-1)\Delta_i^2(\lambda)} = \frac{\lambda \mathbf{E}_i}{\mathbf{S}_i}. \quad (10)$$

30 Now note that  $\sum \mathbf{S}_i e^{(2m-1)\Delta_i^2(\lambda)}$  is an unbounded monotonically increasing function of  $\lambda$ . We now  
31 define  $\lambda^*$  to be the unique nonnegative value such that we have the following.

$$\sum \mathbf{S}_i e^{(2m-1)\Delta_i^2(\lambda^*)} = \frac{4m}{\delta}. \quad (11)$$

32 Note that  $\Delta_i(0) = 0$  and  $\sum \mathbf{S}_i e^{(2m-1)\Delta_i^2(0)} = 1 < \frac{4m}{\delta}$ . So we must have  $\lambda^* > 0$  and hence  
33  $\Delta_i(\lambda^*) > 0$  for  $\mathbf{E}_i > 0$ .  $\square$

34 **Lemma 2.** For any  $\gamma_i$  satisfying Eq. (6), we have the following.

$$\sum \mathbf{E}_i \gamma_i \leq \sum \mathbf{E}_i \Delta_i(\lambda^*). \quad (12)$$

35 *Proof.* Consider the following four situations:

36 (1)  $\exists i \gamma_i < 0$

37  $\sum \mathbf{E}_i \gamma_i$  can be increased by replacing  $\gamma_i$  with  $-\gamma_i$  for  $\gamma_i < 0$ . Hence we can assume without loss of  
38 generality that  $\gamma_i \geq 0$ .

39 (2)  $\sum \mathbf{S}_i e^{(2m-1)\gamma_i^2} < \frac{4m}{\delta}$

40  $\sum \mathbf{E}_i \gamma_i$  can be increased by raising  $\gamma_i$  with  $\mathbf{E}_i > 0$ . Hence we can assume without loss of generality  
41 that  $\sum \mathbf{S}_i e^{(2m-1)\gamma_i^2} = \frac{4m}{\delta}$ .

42 (3)  $\exists i \mathbf{E}_i = 0, \gamma_i > 0$

43  $\sum \mathbf{E}_i \gamma_i$  can be increased by setting  $\gamma_i = 0$  with  $\mathbf{E}_i = 0$  while raising  $\gamma_i$  with  $\mathbf{E}_i > 0$ . Hence we  
44 can assume without loss of generality that  $\gamma_i = 0$  whenever  $\mathbf{E}_i = 0$ .

45 (4)  $\exists j, k \mathbf{E}_j > 0, \mathbf{E}_k > 0,$

46  $\frac{\mathbf{S}_j}{\mathbf{E}_j} \gamma_j e^{(2m-1)\gamma_j^2} > \frac{\mathbf{S}_k}{\mathbf{E}_k} \gamma_k e^{(2m-1)\gamma_k^2}$

Eq. (8) is trivially true if  $\mathbf{E}_i = 0$  and  $\gamma_i = 0$ . So we can rewrite Eq. (8) as follows.

$$\lambda_i = \frac{\mathbf{S}_j}{\mathbf{E}_i} 2(2m-1) \gamma_i e^{(2m-1)\gamma_i^2}. \quad (13)$$

From Eq. (13) we can get the following.

$$\lambda_j = \frac{\mathbf{S}_j}{\mathbf{E}_j} 2(2m-1) \gamma_j e^{(2m-1)\gamma_j^2} > \frac{\mathbf{S}_k}{\mathbf{E}_k} 2(2m-1) \gamma_k e^{(2m-1)\gamma_k^2} = \lambda_k. \quad (14)$$

For  $\lambda_i > 0$ , Eq. (8) can be rewritten as follows.

$$\mathbf{E}_i = \frac{\mathbf{S}_j}{\lambda_i} 2(2m-1) \gamma_i^2 e^{(2m-1)\gamma_i^2}. \quad (15)$$

So we have the following.

$$\sum \mathbf{E}_i \gamma_i = \sum \frac{\mathbf{S}_j}{\lambda_i} 2(2m-1) \gamma_i^2 e^{(2m-1)\gamma_i^2}. \quad (16)$$

Let  $f(x_i)$  be the function mapping  $x_i$  to  $\sum \frac{\mathbf{S}_j}{\lambda_i} 2(2m-1) x_i e^{(2m-1)x_i}$ .

$$f'(x_i) = 4m(2m-1) \sum \frac{\mathbf{S}_i}{\lambda_i} x_i e^{(2m-1)x_i}. \quad (17)$$

For  $x_j = x_k$ ,  $f'(x_j) < f'(x_k)$ .

$\sum \mathbf{E}_i \gamma_i$  can be increased by increasing  $\gamma_k$  and decreasing  $\gamma_j$  while holding  $\sum \mathbf{S}_i e^{(2m-1)\gamma_i^2}$  constant. Hence we can assume without loss of generality that there exists a value  $\lambda'$  such that for all indices  $i$  with  $\mathbf{E}_i > 0$  we have  $2(2m-1) \gamma_i e^{(2m-1)\gamma_i^2} = \frac{\lambda' \mathbf{E}_i}{\mathbf{S}_i}$ , which implies  $\gamma_i = \Delta_i(\lambda')$ . We also have  $\sum \mathbf{S}_i e^{(2m-1)\Delta_i^2(\lambda')} = \frac{4m}{\delta}$ , which implies  $\lambda' = \lambda^*$ . So we have  $\gamma_i = \Delta_i(\lambda^*)$  which implies the result.

Now proving lemma 1 suffices to bound  $\sum \mathbf{E}_i \Delta_i(\lambda^*)$ . Eq. (10) implies that for  $\lambda \gg 1$  and  $\mathbf{E}_i \geq \mathbf{S}_i$  we have the following

$$\Delta_i(\lambda) \approx \sqrt{\frac{\ln \frac{\lambda \mathbf{E}_i}{\mathbf{S}_i}}{2m-1}}. \quad (18)$$

□

This approximate relationship is made more precise in the following two lemmas.

**Lemma 3.** For  $m \geq 1$ ,  $\mathbf{S}_i > 0$ ,  $\mathbf{E}_i \geq \mathbf{S}_i$  and  $\lambda \geq e$ , we have the following.

$$\Delta_i(\lambda) \leq \sqrt{\frac{\ln \frac{\lambda \mathbf{E}_i}{\mathbf{S}_i}}{2m-1}}. \quad (19)$$

*Proof.* Let  $g(x)$  be the function mapping  $x$  to  $2(2m-1)x e^{(2m-1)x^2}$ . By definition,  $\Delta_i(\lambda)$  satisfies  $g(\Delta_i(\lambda)) = \frac{\lambda \mathbf{E}_i}{\mathbf{S}_i}$ . Note that for  $x \geq 0$  we have that  $g(x)$  is a monotonically increasing function. Hence for  $x \geq 0$  and  $g(x) \geq \frac{\lambda \mathbf{E}_i}{\mathbf{S}_i}$  we must have  $\Delta_i(\lambda) \leq x$ . Under the assumptions of lemma 3 we have  $\ln \frac{\lambda \mathbf{E}_i}{\mathbf{S}_i} \geq 1$  which implies the following.

$$g\left(\sqrt{\frac{\ln \frac{\lambda \mathbf{E}_i}{\mathbf{S}_i}}{2m-1}}\right) = 2\sqrt{(2m-1)\ln \frac{\lambda \mathbf{E}_i}{\mathbf{S}_i}} \frac{\lambda \mathbf{E}_i}{\mathbf{S}_i} \geq \frac{\lambda \mathbf{E}_i}{\mathbf{S}_i}. \quad (20)$$

□

**Lemma 4.** For  $m \geq 1$ ,  $\mathbf{S}_i > 0$ ,  $\mathbf{E}_i \geq \mathbf{S}_i$  and  $\lambda \geq e$ , we have the following.

$$\Delta_i(\lambda) \geq \sqrt{\frac{\ln \frac{\lambda \mathbf{E}_i}{\mathbf{S}_i} + \frac{1}{2} \ln m - \ln \ln \frac{\lambda \mathbf{E}_i}{\mathbf{S}_i} - 2}{2m-1}}. \quad (21)$$

69 *Proof.* By an argument similar to that in the proof of lemma 3, to show that  $\Delta_i(\lambda) \geq x$  it suffices to  
 70 show that  $g(x) \leq \frac{\lambda E_i}{S_i}$ . In particular we have the following.

$$\begin{aligned}
 & g\left(\sqrt{\frac{\ln \frac{\lambda E_i}{S_i} + \frac{1}{2} \ln m - \ln \ln \frac{\lambda E_i}{S_i} - 2}{2m-1}}\right) \\
 &= 2\sqrt{(2m-1)\left(\ln \frac{\lambda E_i}{S_i} + \frac{1}{2} \ln m - \ln \ln \frac{\lambda E_i}{S_i} - 2\right)} \\
 &\quad \cdot \frac{\lambda E_i}{S_i} \frac{1}{\sqrt{m} \ln \frac{\lambda E_i}{S_i} e^2} \\
 &\leq 4\sqrt{m \ln \frac{\lambda E_i}{S_i} \frac{\lambda E_i}{S_i} \frac{1}{\sqrt{m} \ln \frac{\lambda E_i}{S_i} e^2}} \\
 &= \frac{\lambda E_i}{S_i} \frac{1}{\sqrt{\ln \frac{\lambda E_i}{S_i} e^2}} \frac{4}{e^2} \\
 &\leq \frac{\lambda E_i}{S_i}.
 \end{aligned} \tag{22}$$

71

□

72 **Lemma 5.** For  $d(E, S) \leq 2m$  and  $\ln \frac{1}{\delta} \leq 2m$ , we have  $\lambda^* \leq \frac{64e^2 m^{5/2}}{\delta}$ .

73 *Proof.* Let  $h(x)$  be the function mapping  $x$  to  $\sum S_i e^{(2m-1)\Delta_i^2(x)}$ . The quantity  $\lambda^*$  is defined by  
 74  $h(\lambda^*) = \frac{4m}{\delta}$ . Since  $h$  is a monotonically increasing function,  $h(x) \geq \frac{2m}{\delta}$  implies  $\lambda^* \leq x$ . Lemma 4  
 75 implies that for  $x \geq e$  we have the following.

$$\begin{aligned}
 h(x) &= \sum S_i e^{(2m-1)\Delta_i^2(x)} \\
 &\geq \sum S_i \frac{x E_i}{S_i} \frac{1}{\sqrt{m} \ln \frac{x E_i}{S_i} e^2} \\
 &= \frac{x}{e^2 \sqrt{m}} \sum \frac{E_i}{\ln \frac{x E_i}{S_i}} \\
 &\geq \frac{x}{e^2 \sqrt{m}} \frac{1}{\sum E_i \ln \frac{x E_i}{S_i}} \\
 &= \frac{x}{e^2 \sqrt{m} (d(E, B) + \ln x)} \\
 &\geq \frac{x}{e^2 \sqrt{m} (2m + \ln x)}.
 \end{aligned} \tag{23}$$

76 Now inserting  $x = \frac{64e^2 m^{5/2}}{\delta}$  we get the following.

$$\begin{aligned}
 h\left(\frac{64e^2 m^{5/2}}{\delta}\right) &\geq \frac{64m^2}{\delta(2m + \frac{5}{2} \ln m + \ln \frac{1}{\delta} + 8)} \\
 &\geq \frac{64m^2}{\delta(2m + \frac{5}{2}m + 2m + 8m)} \\
 &\geq \frac{4m}{\delta}.
 \end{aligned} \tag{24}$$

77

□

78 **Lemma 6.** Without loss of generality, we define the distance between distributions as  $d(P, Q) =$   
 79  $\sum P_i \ln \frac{P_i}{Q_i}$  for analysis. For  $\ln \frac{1}{\delta} \leq 2m$  we have that with probability  $1 - \delta$  over the training dataset  
 80 of size  $m$  the following holds for all distributions satisfying  $d(E, S) \leq 2m$ .

$$d(E, O) - d(E, S) \leq \sum E_i \sqrt{\frac{\ln \frac{E_i}{S_i} + \ln \frac{1}{\delta} + \frac{5}{2} \ln m + 8}{2m-1}}. \tag{25}$$

81 *Proof.* Note that  $d(\mathbf{E}, \mathbf{O}) - d(\mathbf{E}, \mathbf{S}) = \sum \mathbf{E}_i (\ln \frac{\mathbf{E}_i}{\mathbf{O}_i} - \ln \frac{\mathbf{E}_i}{\mathbf{S}_i}) \leq \sum \mathbf{E}_i \gamma_i$ , where  $\gamma_i$  abbreviates  
 82  $|\ln \frac{\mathbf{E}_i}{\mathbf{O}_i} - \ln \frac{\mathbf{E}_i}{\mathbf{S}_i}|$ , the lemma can be viewed as an upper bound of  $\sum \mathbf{E}_i \gamma_i$ .  
 83 From the above-mentioned lemmas, we have that

$$\begin{aligned}
 d(\mathbf{E}, \mathbf{O}) - d(\mathbf{E}, \mathbf{S}) &\leq \sum \mathbf{E}_i \gamma_i \\
 &\leq \sum \mathbf{E}_i \Delta_i(\lambda^*) \\
 &\leq \sum \mathbf{E}_i \Delta_i\left(\frac{64e^2 m^{5/2}}{\delta}\right) \\
 &\leq \sum \mathbf{E}_i \sqrt{\frac{\ln \frac{64e^2 m^{5/2} \mathbf{E}_i}{\delta \mathbf{S}_i}}{2m-1}} \\
 &\leq \sum \mathbf{E}_i \sqrt{\frac{\ln \frac{\mathbf{E}_i}{\mathbf{S}_i} + \ln \frac{1}{\delta} + \frac{5}{2} \ln m + 8}{2m-1}}.
 \end{aligned} \tag{26}$$

84 □

85 **Lemma 7.** Denote the  $d(\cdot, \cdot)$  as the distribution distance. With probability at least  $1 - \delta$ , we have  
 86 the following,

$$\begin{aligned}
 d(p(\mathbf{x}_o, \mathbf{y}_o), p(\mathbf{x}_e, \mathbf{y}_e)) &\leq d(p(\mathbf{x}_e, \mathbf{y}_e), p(\mathbf{x}_s, \mathbf{y}_s)) + \sqrt{\frac{d(p(\mathbf{x}_e, \mathbf{y}_e), p(\mathbf{x}_s, \mathbf{y}_s))}{2m-1}} \\
 &\quad + \sqrt{\frac{\ln \frac{1}{\delta} + \frac{5}{2} \ln m + 8}{2m-1}},
 \end{aligned} \tag{27}$$

87 where  $m$  denotes the size of training dataset.

88 *Proof.* By applying Jensen's inequality on lemma 6, we can get Theorem 1.

$$\begin{aligned}
 d(\mathbf{E}, \mathbf{O}) &\leq d(\mathbf{E}, \mathbf{S}) + \sum \mathbf{E}_i \sqrt{\frac{\ln \frac{\mathbf{E}_i}{\mathbf{S}_i} + \ln \frac{1}{\delta} + \frac{5}{2} \ln m + 8}{2m-1}} \\
 &\leq d(\mathbf{E}, \mathbf{S}) + \sqrt{\frac{d(\mathbf{E}, \mathbf{S}) + \ln \frac{1}{\delta} + \frac{5}{2} \ln m + 8}{2m-1}} \\
 &\leq d(\mathbf{E}, \mathbf{S}) + \sqrt{\frac{d(\mathbf{E}, \mathbf{S})}{2m-1}} + \sqrt{\frac{\ln \frac{1}{\delta} + \frac{5}{2} \ln m + 8}{2m-1}}.
 \end{aligned} \tag{28}$$

89 □

90 From Lemma 7, we can obviously observe that the distribution distance of the generated unseen-class  
 91 data and the open-environment has an upper bound, which indicates that the rationality of generating  
 92 unseen-class data for distribution estimation in open environments. Obviously, we also can observe  
 93 that this upper bound is strongly related to the distribution distance between the generated unseen-  
 94 class data and the seen-class data. The conclusion of Lemma 7 is same with Theorem 1. This also  
 95 motivates us to construct the proposed open-vocabulary method. From Lemma 7, we can directly  
 96 obtain Theorem 1.

97 **Theorem 1.** Denote the  $d(\cdot, \cdot)$  as the distribution distance. With probability at least  $1 - \delta$ , we have  
 98 the following,

$$\begin{aligned}
 d(p(\mathbf{x}_u, \mathbf{y}_u), p(\mathbf{x}_e, \mathbf{y}_e)) &\leq \sqrt{\frac{d(p(\mathbf{x}_e, \mathbf{y}_e), p(\mathbf{x}_s, \mathbf{y}_s))}{2m-1}} \\
 &\quad + \sqrt{\frac{\ln \frac{1}{\delta} + \frac{5}{2} \ln m + 8}{2m-1}},
 \end{aligned} \tag{29}$$

99 where  $m$  denotes the size of training dataset.

## 2 Proof of Theorem 2

In the manuscripts, we present Theorem 2 to demonstrate that the estimation error of  $p(\bar{\mathbf{y}}|\mathbf{x}_u, \Phi)$  is upper bounded. Here we provide specific derivations of Theorem 2.

**Theorem 2.** *Given the predicted classes  $\mathbf{Y}_e = \{\mathbf{y}_e\}$ . Suppose that predicted class  $\mathbf{Y}_e$  have any nonzero probability  $p(\mathbf{Y}_e)$ . With probability at least  $1 - \delta$  over the  $m$  instances of generated unseen-class data  $\{(\mathbf{x}_e, \mathbf{y}_e)\}$ , we have that*

$$D_{\text{KL}}(p(\bar{\mathbf{y}}|\mathbf{x}_e, \Phi) || p(\bar{\mathbf{y}}|\mathbf{x}_u, \Phi)) \leq \sqrt{\frac{\ln \frac{1}{p(\mathbf{Y}_e)} + \ln \frac{1}{\delta}}{2m}} - \ln \frac{|\mathbf{Y}_e|}{|\mathbf{Y}_u|}. \quad (30)$$

*Proof.* We denote  $p^*(\mathbf{y}) = p(\bar{\mathbf{y}}|\mathbf{x}_u, \Phi)$ ,  $\bar{\mathbf{y}} \in \mathbf{Y}_u$ . For analysis, we define  $P(\mathbf{Y}_e) = \sum_{\mathbf{y} \in \mathbf{Y}_e} p^*(\mathbf{y})$ , and we define the conditional distribution as  $P_e(\mathbf{y}) = p(\bar{\mathbf{y}}|\mathbf{x}_u, \Phi) = \frac{p^*(\mathbf{y})}{P(\mathbf{Y}_e)}$ ,  $\bar{\mathbf{y}} \in \mathbf{Y}_e$ .  $P_e(\mathbf{y})$  satisfies that  $\sum_{\mathbf{y} \in \mathbf{Y}_e} P_e(\mathbf{y}) = 1$ . We also denote that  $\hat{p}(\mathbf{y}) = p(\bar{\mathbf{y}}|\mathbf{x}_e, \Phi)$ . In this way, the KL divergence between  $\hat{p}(\mathbf{y})$  and  $p^*(\mathbf{y})$  is computed as

$$D_{\text{KL}}(\hat{p}(\mathbf{y}) || p^*(\mathbf{y})) = \sum_{\mathbf{y} \in \mathbf{Y}_e} \hat{p}(\mathbf{y}) \ln \frac{\hat{p}(\mathbf{y})}{p^*(\mathbf{y})}. \quad (31)$$

$\frac{\hat{p}(\mathbf{y})}{p^*(\mathbf{y})}$  can be formulated as

$$\frac{\hat{p}(\mathbf{y})}{p^*(\mathbf{y})} = \frac{\hat{p}(\mathbf{y})}{P_e(\mathbf{y})} \frac{P_e(\mathbf{y})}{p^*(\mathbf{y})} = \frac{\hat{p}(\mathbf{y})}{P_e(\mathbf{y})} \frac{1}{P(\mathbf{Y}_e)}, \quad (32)$$

and thus

$$\ln \frac{\hat{p}(\mathbf{y})}{p^*(\mathbf{y})} = \ln \frac{\hat{p}(\mathbf{y})}{P_e(\mathbf{y})} - \ln P(\mathbf{Y}_e). \quad (33)$$

By substituting Eq. (33) into Eq (31), we have that

$$\begin{aligned} D_{\text{KL}}(\hat{p}(\mathbf{y}) || p^*(\mathbf{y})) &= \sum_{\mathbf{y} \in \mathbf{Y}_e} \hat{p}(\mathbf{y}) \left( \ln \frac{\hat{p}(\mathbf{y})}{P_e(\mathbf{y})} - \ln P(\mathbf{Y}_e) \right) \\ &= \sum_{\mathbf{y} \in \mathbf{Y}_e} \hat{p}(\mathbf{y}) \ln \frac{\hat{p}(\mathbf{y})}{P_e(\mathbf{y})} - \sum_{\mathbf{y} \in \mathbf{Y}_e} \hat{p}(\mathbf{y}) \ln P(\mathbf{Y}_e) \\ &= D_{\text{KL}}(\hat{p}(\mathbf{y}) || P_e(\mathbf{y})) - \ln P(\mathbf{Y}_e). \end{aligned} \quad (34)$$

By the Chernoff bound, for the generated unseen-class data  $\{(\mathbf{x}_e, \mathbf{y}_e)\}$ , we have that

$$P[D_{\text{KL}}(\hat{p}(\mathbf{y}) || P_e(\mathbf{y})) \geq \delta] \leq e^{-2mt^2}. \quad (35)$$

From the union bound, and the classes are countable, we have that

$$P[\exists \mathbf{Y}_e : D_{\text{KL}}(\hat{p}(\mathbf{y}) || P_e(\mathbf{y})) \geq t] \leq \sum_{\mathbf{Y}_e} P[D_{\text{KL}}(\hat{p}(\mathbf{y}) || P_e(\mathbf{y})) \geq t]. \quad (36)$$

Assign the probability  $p(\mathbf{Y}_e)$  for  $\mathbf{Y}_e$ , and thus  $\sum_{\mathbf{Y}_e} p(\mathbf{Y}_e) = 1$ . In this way, we have that

$$P[\exists \mathbf{Y}_e : D_{\text{KL}}(\hat{p}(\mathbf{y}) || P_e(\mathbf{y})) \geq \delta] \leq \sum_{\mathbf{Y}_e} e^{-2mt^2} = \sum_{\mathbf{Y}_e} p(\mathbf{Y}_e) \frac{e^{-2mt^2}}{p(\mathbf{Y}_e)}. \quad (37)$$

Letting  $e^{-2mt^2} = p(\mathbf{Y}_e)\delta$ , we have that

$$P[\exists \mathbf{Y}_e : D_{\text{KL}}(\hat{p}(\mathbf{y}) || P_e(\mathbf{y})) \geq t] \leq \sum_{\mathbf{Y}_e} p(\mathbf{Y}_e)\delta = \delta, \quad (38)$$

where  $t$  is computed as

$$t = \sqrt{\frac{\ln \frac{1}{p(\mathbf{Y}_e)} + \ln \frac{1}{\delta}}{2m}}. \quad (39)$$

118 In this way, we can derive that

$$P \left[ D_{\text{KL}}(\hat{p}(\mathbf{y}) || P_e(\mathbf{y})) \geq \sqrt{\frac{\ln \frac{1}{p(\mathbf{y}_e^i)} + \ln \frac{1}{\delta}}{2m}} \right] \leq \delta, \quad (40)$$

119 Therefore, with probability at least  $1 - \delta$ , we have that

$$D_{\text{KL}}(\hat{p}(\mathbf{y}) || P_e(\mathbf{y})) \leq \sqrt{\frac{\ln \frac{1}{p(\mathbf{y}_e^i)} + \ln \frac{1}{\delta}}{2m}}. \quad (41)$$

120 We substitute Eq. (41) into Eq. (34). With probability at least  $1 - \delta$ , we have that

$$D_{\text{KL}}(p(\bar{\mathbf{y}}|\mathbf{x}_e, \Phi) || p(\bar{\mathbf{y}}|\mathbf{x}_u, \Phi)) \leq \sqrt{\frac{\ln \frac{1}{p(\mathbf{Y}_e)} + \ln \frac{1}{\delta}}{2m}} - \ln \frac{|\mathbf{Y}_e|}{|\mathbf{Y}_u|}. \quad (42)$$

121

□

### 122 3 Proof of ELBO in Eq. (7) in the Manuscripts

123 **Proposition 1.** *The Evidence Lower Bound (ELBO) of the logarithmic posterior probability can be*  
 124 *derived as*

$$\log p(\bar{\mathbf{y}}|\mathbf{x}_o, \Phi) \geq \mathbb{E}[\log p(\bar{\mathbf{y}}|\mathbf{x}_s, \Phi)] - D_{\text{KL}}(p(\bar{\mathbf{y}}|\mathbf{x}_s, \Phi) || p(\bar{\mathbf{y}}|\mathbf{x}_u, \Phi)), \quad (43)$$

125 *Proof.* The probability  $p(\bar{\mathbf{y}}|\mathbf{x}_o)$  can be modeled as a function  $f(\mathbf{y})$ . Therefore, the logarithmic  
 126 probability  $\log p(\bar{\mathbf{y}}|\mathbf{x}_o)$  is computed as

$$\log p(\bar{\mathbf{y}}|\mathbf{x}_o) = \log f(\mathbf{y}) = \log \left( \int f(\tilde{\mathbf{y}}) \delta(\tilde{\mathbf{y}} - \mathbf{y}) d\tilde{\mathbf{y}} \right), \quad (44)$$

127 where  $\delta(\cdot)$  is a Dirac function, and  $\tilde{\mathbf{y}}$  is a intermediate variable. The last equality holds since the  
 128 proposition of the Dirac function. We introduce a variational distribution  $q(\tilde{\mathbf{y}})$ . Then,  $\log f(\mathbf{y})$  holds  
 129 that

$$\begin{aligned} \log f(\mathbf{y}) &= \log \left( \int f(\tilde{\mathbf{y}}) \delta(\tilde{\mathbf{y}} - \mathbf{y}) d\tilde{\mathbf{y}} \right) = \log \left( \int f(\tilde{\mathbf{y}}) \frac{q(\tilde{\mathbf{y}})}{q(\tilde{\mathbf{y}})} \delta(\tilde{\mathbf{y}} - \mathbf{y}) d\tilde{\mathbf{y}} \right) \\ &= \log \left( \int q(\tilde{\mathbf{y}}) \frac{f(\tilde{\mathbf{y}}) \delta(\tilde{\mathbf{y}} - \mathbf{y})}{q(\tilde{\mathbf{y}})} d\tilde{\mathbf{y}} \right). \end{aligned} \quad (45)$$

130 From the Jensen inequality, we can derive that

$$\log f(\mathbf{y}) \geq \int q(\tilde{\mathbf{y}}) \log \frac{f(\tilde{\mathbf{y}}) \delta(\tilde{\mathbf{y}} - \mathbf{y})}{q(\tilde{\mathbf{y}})} d\tilde{\mathbf{y}} = \mathbb{E}_{q(\mathbf{y})} \left[ \log \frac{f(\mathbf{y}) \delta(\tilde{\mathbf{y}} - \mathbf{y})}{q(\mathbf{y})} \right]. \quad (46)$$

131 To guarantee the boundedness of the ELBO, we ignore the Dirac function. Eq. (46) can be further  
 132 modeled as

$$\log f(\mathbf{y}) \geq \mathbb{E}_{q(\mathbf{y})} [\log f(\mathbf{y})] - \mathbb{E}_{q(\mathbf{y})} [\log q(\mathbf{y})]. \quad (47)$$

133 Then, substituting  $f(\mathbf{y}) = \log p(\bar{\mathbf{y}}|\mathbf{x}_o, \Phi)$  and  $q(\mathbf{y}) = p(\bar{\mathbf{y}}|\mathbf{x}_s, \Phi)$ , we have that

$$\begin{aligned} \log p(\bar{\mathbf{y}}|\mathbf{x}_o, \Phi) &\geq \mathbb{E}_{p(\bar{\mathbf{y}}|\mathbf{x}_s, \Phi)} [\log p(\bar{\mathbf{y}}|\mathbf{x}_s, \Phi)] + \mathbb{E}_{p(\bar{\mathbf{y}}|\mathbf{x}_s, \Phi)} [\log p(\bar{\mathbf{y}}|\mathbf{x}_u, \Phi)] \\ &\quad - \mathbb{E}_{p(\bar{\mathbf{y}}|\mathbf{x}_s, \Phi)} [\log p(\bar{\mathbf{y}}|\Phi)] - \mathbb{E}_{p(\bar{\mathbf{y}}|\mathbf{x}_s, \Phi)} [\log p(\bar{\mathbf{y}}|\mathbf{x}_s, \Phi)] \\ &= \mathbb{E}_{p(\bar{\mathbf{y}}|\mathbf{x}_s, \Phi)} [\log p(\bar{\mathbf{y}}|\mathbf{x}_s, \Phi)] + \mathbb{E}_{p(\bar{\mathbf{y}}|\mathbf{x}_s, \Phi)} \left[ \log \frac{p(\bar{\mathbf{y}}|\mathbf{x}_u, \Phi)}{p(\bar{\mathbf{y}}|\mathbf{x}_s, \Phi)} \right] \\ &\quad - \mathbb{E}_{p(\bar{\mathbf{y}}|\mathbf{x}_s, \Phi)} [\log p(\bar{\mathbf{y}}|\Phi)]. \end{aligned} \quad (48)$$

134 The probability term  $p(\bar{\mathbf{y}}|\Phi)$  can be ignored because it appears as a constant term in the variational  
 135 lower bound (ELBO). It depends only on the model parameters  $\Phi$  and is independent of the input  
 136 data  $\mathbf{x}$ . As such, it does not affect the gradient computation or the update of model parameters  $\Phi$ .  
 137 Since this term does not contribute to the optimization process, it can be safely omitted, simplifying  
 138 the derivation. In many works, derivation of ELBO commonly omit such constant terms, as they do  
 139 not affect the optimization objective and can be safely ignored to simplify the computation [3, 1, 2].  
 140 Therefore, we model Eq. (48) as

$$\log p(\bar{\mathbf{y}}|\mathbf{x}_o, \Phi) \geq \mathbb{E} [\log p(\bar{\mathbf{y}}|\mathbf{x}_s, \Phi)] - D_{\text{KL}}(p(\bar{\mathbf{y}}|\mathbf{x}_s, \Phi) || p(\bar{\mathbf{y}}|\mathbf{x}_u, \Phi)). \quad (49)$$

141

□

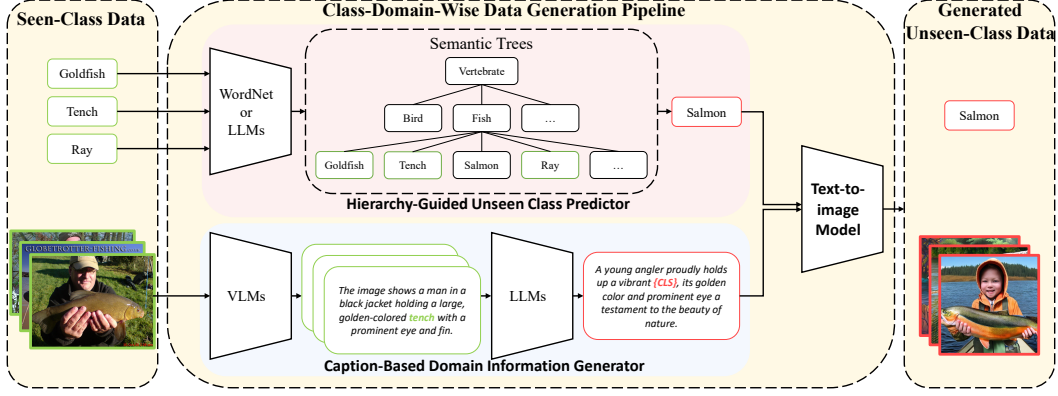


Figure 1: Formulation of Class-Domain-Wise Data Generation Pipeline

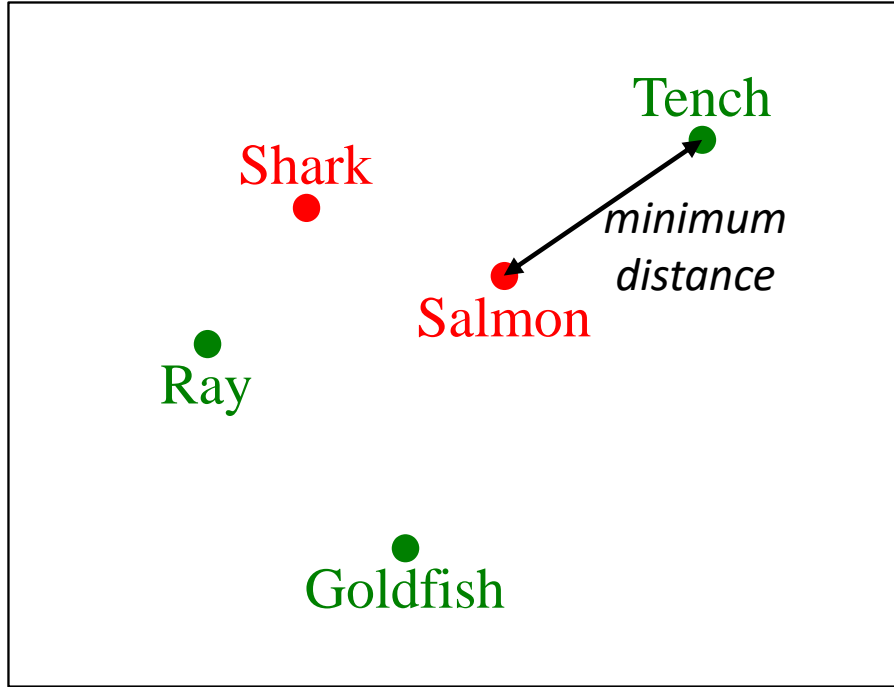


Figure 2: Distance between Classes

#### 142 4 Details and Illustration of Class-Domain-Wise Data Generation Pipeline

143 In the manuscripts, we propose an class-domain-wise data generation pipeline to generate unseen-class  
 144 data, which contains an unseen-class predictor, domain information generator, and a text-to-image  
 145 model. Here, we illustrate the proposed pipeline, as shown in Figure 1. Moreover, we present the  
 146 specific details and examples for better understanding.

##### 147 4.1 Specific Details of Hierarchy-Guided Unseen Class Predictor

148 The hierarchy-guided unseen class predictor identifies the potential unseen classes, which are close  
 149 to training classes. This is achieved by constructing a hierarchical semantic tree, where leaf nodes  
 150 represent training classes and parent nodes represent their superclasses. The tree is expanded by  
 151 adding leaf nodes of the candidate unseen classes sourced from WordNet or LLMs. As illustrated in  
 152 Figure 1, given the training classes “Goldfish,” “Tench,” and “Ray,” a hierarchical semantic tree is  
 153 constructed where these classes are set as leaf nodes and their superclass “Fish” is set as a parent



node. After LLMs are queried, "Salmon" is added as a leaf node of the candidate unseen class under the "Fish" superclass. To select the closest candidate unseen class, the predictor computes the cosine similarity between the textual embeddings of candidate unseen classes and given training classes from the text encoder of a pretrained CLIP. The top  $K$  closest candidates are chosen as the identified potential unseen classes. Take the classes shown in Figure 2 as an example, the cosine similarity between the textual embeddings of "Tench" and candidate unseen classes "Salmon", "Shark" is computed. The candidate unseen class with the highest similarity "Salmon" is chosen as a predicted unseen class.

## 4.2 Specific Details of Caption-Based Domain Information Generator

The caption-based domain information generator extracts contextual attributes, such as styles and backgrounds, from the training data to ensure the generated unseen data align with the visual characteristics of training data. This is achieved by generating class-specific captions for each training class using VLMs. The generator then computes the similarity between these captions and the corresponding data, selecting the top  $K_1$  captions with the highest similarity to mitigate hallucination issues. These selected captions are further summarized into top  $K_2$  class-specific domain information using LLMs. Finally, the predicted unseen classes and the summarized domain information are combined into textual prompts, which guide the data generation process using a text-to-image model such as Stable Diffusion. In Figure 1, for instance, captions for training images of "Tench" are first generated using VLMs. The generator then calculates the similarity between these captions and the corresponding images of "Tench". The top 3 captions with the highest similarity are selected, which describe the domain information such as "holded by man", "golden-colored" and "prominent eye". These captions are summarized into top 1 class-specific domain information using LLMs, which is presented in the caption template in the red box. Finally, the predicted unseen class "Salmon" is inserted into the template to create an image caption, which is then used as input for Stable Diffusion to generate images of "Salmon".

## 4.3 Prompt Templates

In the manuscripts, we propose to utilize LLMs and VLMs to identify potential unseen classes and extract domain information of training data. Here we provide the utilized 3 prompt templates.

As to the unseen class predictor, we aim to query the Hypernym of training classes by the following prompt. We first construct the in-context examples for accurate results, as shown in Template 1.

**Template 1.** *{class} denotes the training class.*  
*Q: What is the Hypernym category of {class1}?*  
*A: {class2} is the Hypernym category of {class1}.*

The prompt is given in Template 2.

**Template 2.** *{class} denotes the training class.*  
*Q: What is the Hypernym category of {class3}?*

Then, with the generated Hypernym, we leverage LLMs to identify potential unseen classes using LLMs, where the in-context examples and prompts are shown in template 3 and template 4, respectively.

**Template 3.** *{class} denotes the training class.*  
*Q: What is the Hyponym category of {class1}?*  
*A: {class2} is the Hyponym category of {class1}.*

**Template 4.** *{class} denotes the training class.*  
*Q: What is the Hyponym category of {class3}?*

We leverage template 5 to generate class-specific captions for each training class using VLMs.

**Template 5.** *{class} denotes the training class.*  
*user prompt:*

205 *This is an image of {class}. Summarize the main style, scene, and key elements of this image in one*  
 206 *sentence.*

207 We leverage template 6 to summarize captions into class-specific domain information using LLMs

208 **Template 6.** *{class}* denotes the predicted unseen class.

209 *system prompt:*

210 *As a caption summarizer, your task is to transform the provided captions from their original category*  
 211 *to a new specified category and condense them into a concise set of 3 distinct one-sentence captions.*  
 212 *Make sure the new captions maintain coherence with the original style but reflect the characteristics*  
 213 *of the new target category. Each caption must capture a unique artistic style or visual theme. Only*  
 214 *generate the transformed one-sentence captions—no introductions, explanations, or comments. The*  
 215 *output should strictly follow this format:*

216 *1. [Caption 1]*

217 *2. [Caption 2]*

218 *3. [Caption 3]*

219 *user prompt:*

220 *Transform and condense the following captions into 3 new one-sentence captions describing {class},*  
 221 *each focusing on a distinct artistic style or visual theme.*

---

**Algorithm 1** Distribution Alignment Algorithm

---

**Input:** Parameters  $\Phi$ , Data  $\mathcal{D}_s, \mathcal{G}_u, \mathcal{G}_s$ , Epoch  $E$ , batch-size  $B$

**Output:** Optimized Prompts  $v_{max\_iter}$

**Initialize:**  $e = 0, v \leftarrow v_0, S \leftarrow \{\}$

```

1: while  $e \leq E$  do
2:   for  $i = 1, 2, \dots, |\mathcal{D}_s|/B$  do
3:     Compute the posterior probability of model output in the current batch of the seen-class
       dataset  $\mathcal{D}_s^i$ .
4:     Accumulate the output posterior probability for distribution alignment into  $S$ .
5:     if  $i \% 8 == 0$  then
6:        $S_{KL} = \{\}$ .
7:       for  $j = 1, 2, \dots, |\mathcal{G}_u|/B$  do
8:         Compute the KL divergence between the accumulated posterior probability on
           the seen-class data and the mini-batch of generated unseen-class data  $d_{kl} =$ 
            $D_{KL}[p(\bar{y}|\mathbf{x}_s, \Phi, v)||p(\bar{y}|\mathbf{x}_e, \Phi, v)]$ .
9:         Update the set as  $S_{KL}.append(d_{kl})$ .
10:      end for
11:      Compute  $\mathcal{L}_{KL}$  based on top  $K_3$  smallest in the set  $S_{KL}$  as  $\mathcal{L}_{KL} = \frac{1}{K_3} \sum_{topK_3} d_{kl}$ .
12:       $S_{mmd} = \{\}$ .
13:      for  $m = 1, 2, \dots, |\mathcal{G}_s|/B$  do
14:        Compute MMD loss  $l_{mmd}$  based on the generated unseen-class data and the seen-class
          data on the current batch, and save them into  $S_{mmd}$ .
15:      end for
16:      Compute  $\mathcal{L}_{MMD}$  based on top  $K_3$  smallest  $l_{mmd}$  as  $\mathcal{L}_{MMD} = \frac{1}{K_3} \sum_{topK_3} l_{mmd}$ .
17:      Compute total loss  $\mathcal{L}_{total} = \mathcal{L}_{CE} + \alpha \mathcal{L}_{KL} + \beta \mathcal{L}_{MMD}$ .
18:      Backward and update the prompt  $v$  using  $\mathcal{L}_{CE}$  and  $\mathcal{L}_{total}$ .
19:      Clear saved data  $S = \{\}$ .
20:    else
21:      Compute  $\mathcal{L}_{CE}$  on the mini-batch of seen-class dataset  $\mathcal{D}_s^i$ .
22:      Backward and update the prompt  $v$  using  $\mathcal{L}_{CE}$ .
23:    end if
24:  end for
25: end while
26: return The updated prompts  $v$ .

```

---

## 5 Distribution Alignment

In this paper, we adopt prompt learning method to optimize the pretrained model. We propose a distribution alignment algorithm which aligns the output distributions of model on seen-class data and generated unseen-class data to maximize the logarithmic posterior probability in open environments. The proposed algorithm is summarized in Algorithm 1.

## 6 Experiment Details in Manuscripts

In this section, we provide more specific details of experiments in the manuscripts. Specifically, we present the specific implementation details, details and extra analysis in ablation studies of the manuscripts.

### 6.1 Implementation details

For base-to-base/base-to-new generalization, we train each model for 20 epochs using 4 token prompts in the first 9 transformer layers on both visual and text branch. For cross-dataset evaluation, we train the source model for 4 epochs using 4 prompts in the first 3 transformer layers on both visual and text branch. Prompts are randomly initialized with a normal distribution except the text prompts of the first layer which are initialized with the word embeddings of “a photo of a”. The SGD optimizer is adopted, and the learning rate is set as 0.0025. Hyperparameters for the class-domain-wise data generation pipeline and distribution alignment are determined empirically. Specifically, We set  $\alpha = 1$ ,  $\beta = 1$ ,  $K_0$  as 1,  $K_1$  as 8,  $K_2$  as 3 and  $K_3$  as 1. The corresponding hyperparameters are fixed across all datasets and benchmarks.

For LLMs and VLMs, we use Doubao-pro-128k to identifies the potential unseen classes, use LLaVA-v1.6-Vicuna-13B [4] to generate class-specific captions for each training class, use Llama-v3.1-Instruct-8B [8] to summarize captions into class-specific domain information, and use Stable Diffusion v2.1 [7] as the text-to-image model to generate unseen-class data.

Experiments are performed on an NVIDIA A40 GPU, with at most 18 hours 20 GPU memory required to complete training across 11 datasets.

### 6.2 Details and Extra Analysis of Ablation Study

In this subsection, we present the details in ablation studies in hierarchy-guided unseen class predictor and caption-based domain information generator. Then, we present the extra analysis in ablation studies of the manuscripts.

#### 6.2.1 Details of Hierarchy-Guided Unseen Class Predictor

To evaluate the effectiveness of hierarchy-guided unseen class predictor, we first investigate how the quantity of predicted unseen classes impacts the model performance in open environments. Specifically, we introduce a class sampling ratio  $s_{cls}$  to control the quantity of predicted unseen classes used for training. Assume that we initially generate  $N$  unseen classes for a given dataset, with each class containing  $M$  images, we randomly select  $s_{cls} \times N$  classes from the predicted  $N$  unseen classes and then train the model using  $s_{cls} \times N \times M$  images of these classes, discarding the remaining classes and their images.

Next, we investigate how the quality of predicted unseen classes impacts the model performance in open environments. “Low Similarity” denotes that when predicting unseen classes, we compute the cosine similarity to textual seen classes for each candidate class and choose the one with the lowest cosine similarity as the predicted unseen class. “w/oTree” denotes that instead of constructing a hierarchical semantic tree to predict unseen classes, we directly ask LLMs to provide a predicted unseen class corresponding to the given base classes.

#### 6.2.2 Details of Caption-Based Domain Information Generator

To evaluate the effectiveness of caption-based domain information generator, we first investigate how the quantity of generated unseen-class images impacts the model performance in open environments.

Table 1: Ablation study on sparse loss computation strategy. “w/o spa” denotes distribution alignment algorithm without sparse loss computation strategy.

	Caltech		Pets		Cars		Flowers		Food		Aircraft		DTD		EuroSAT		UCF	
	w/o spa	Ours	w/o spa	Ours	w/o spa	Ours	w/o spa	Ours	w/o spa	Ours	w/o spa	Ours	w/o spa	Ours	w/o spa	Ours	w/o spa	Ours
Base	98.52	<b>98.97</b>	93.36	<b>96.01</b>	81.88	<b>82.93</b>	96.68	<b>98.77</b>	91.14	<b>91.39</b>	46.40	<b>48.98</b>	82.41	<b>85.53</b>	95.95	<b>97.17</b>	87.07	<b>89.14</b>
New	94.21	<b>95.85</b>	91.33	<b>97.65</b>	79.48	<b>80.81</b>	78.09	<b>80.92</b>	92.56	<b>92.99</b>	39.71	<b>44.03</b>	61.72	<b>71.50</b>	76.92	<b>87.90</b>	79.61	<b>82.53</b>
H	96.32	<b>97.38</b>	92.33	<b>96.82</b>	80.66	<b>81.86</b>	86.39	<b>88.96</b>	91.84	<b>92.18</b>	42.80	<b>46.37</b>	70.58	<b>77.89</b>	85.39	<b>92.30</b>	83.17	<b>85.71</b>

Similarly, we introduce a image sampling ratio  $s_{img}$  to control the number of generated images of predicted unseen classes used for training. For a given dataset, we initially predict  $N$  unseen classes with each class containing  $M$  generated images. Based on the image sampling ratio  $s_{img}$ , we randomly select  $s_{img} \times M$  images from each predicted unseen class. The selected  $N \times s_{img} \times M$  images from  $N$  unseen classes are then used for training, while the remaining images are discarded.

Next, we investigate how the quality of generated images from predicted unseen classes impacts the model performance in open environments. We modify the prompts used in our unseen image generator to control the quality of image generation. In this work, we use the prompts “A picture of a category”, “A photo of a category”, and “An image of a category” as templates for the Stable Diffusion model when generating images of unseen classes, respectively.

### 6.3 Extra Analysis

The ablation studies provide a comprehensive analysis of the relationship between distribution distance and accuracy, which aligns with the theoretical analysis. Recall that Theorem 1 shows that the estimation error between the unseen-class data distribution and the generated unseen-class data distribution is upper-bounded, and reducing the distribution gap between generated unseen-class data and seen-class data tightens this bound, thereby improving model performance in open environments. Experimental results from the ablation studies validate this theoretical claim. Specifically, as the distribution distance decreases, accuracy consistently improves across various datasets, particularly for new classes. This confirms that reducing the distribution gap between generated unseen-class and seen-class data leads to more accurate estimation of unseen-class data distribution, enhancing the model’s generalization ability in open-vocabulary learning tasks.

## 7 Extra Ablation Studies

In this section, we present extra ablation studies for validating the effectiveness of sparse loss computation strategy in distribution alignment algorithm. We demonstrate the effectiveness of the sparse loss computation strategy by conducting experiments on the distribution alignment algorithm without it, denoted as ‘w/o spa’. The results, shown in Table 1, reveal that the sparse loss computation strategy significantly improves performance, particularly on the new classes. Notably, on the Pets, Cars, DTD, and EuroSAT datasets, the strategy achieves improvements of 6.32%, 9.18%, 9.78%, and 10.98% on the new classes compared to ‘w/o spa’. These results further confirm the effectiveness of the proposed strategy.

## 8 Visualization

In this section, we visualize the unseen-class images generated to demonstrate the effectiveness of the proposed class-domain-wise data generation pipeline. We compare the proposed method with three prompt templates for the text-to-image model mentioned in the ablation studies, namely, ‘A picture of a class’, ‘A photo of class’, and ‘An image of a class’. We use the images generated based on the Caltech101 dataset for analysis.

We use the caption-based domain information generator to capture the class-specific domain information of seen-class data. This domain information is then used to generate the corresponding seen-class data via a text-to-image model. For visualization, we adopt the seen classes ‘motorbike’ and ‘barrel’. As shown in Figures 3 and 4, compared to the three commonly used prompt templates, the generated seen-class data from the proposed pipeline better align with the seen-class data in terms of both style

309 and scene information. This demonstrates that our pipeline is more effective at capturing the domain  
310 information of seen-class images for data generation.

311 Regarding the generation of unseen-class data, we use the hierarchy-based unseen class predictor to  
312 infer that the unseen classes ‘car’ and ‘drum’ are closest to ‘motorbike’ and ‘barrel’, respectively.  
313 The captured class-specific domain information and inferred unseen classes are then used to generate  
314 the unseen-class images via a text-to-image model. We compare the generated unseen-class data  
315 from our pipeline with data generated using the three commonly used prompt templates. As shown in  
316 Figures 3 and 4, the generated unseen-class data align better with the seen-class data. For example, in  
317 Figure 3, the car generated by our pipeline reflects the style of the seen-class data, and the realistic  
318 scene depicted in the generated images mirrors the scene in the seen-class data. These results further  
319 demonstrate that our pipeline effectively captures the domain information of seen-class images, and  
320 the generated unseen-class images align closely with seen-class data, confirming the effectiveness of  
321 the proposed pipeline.

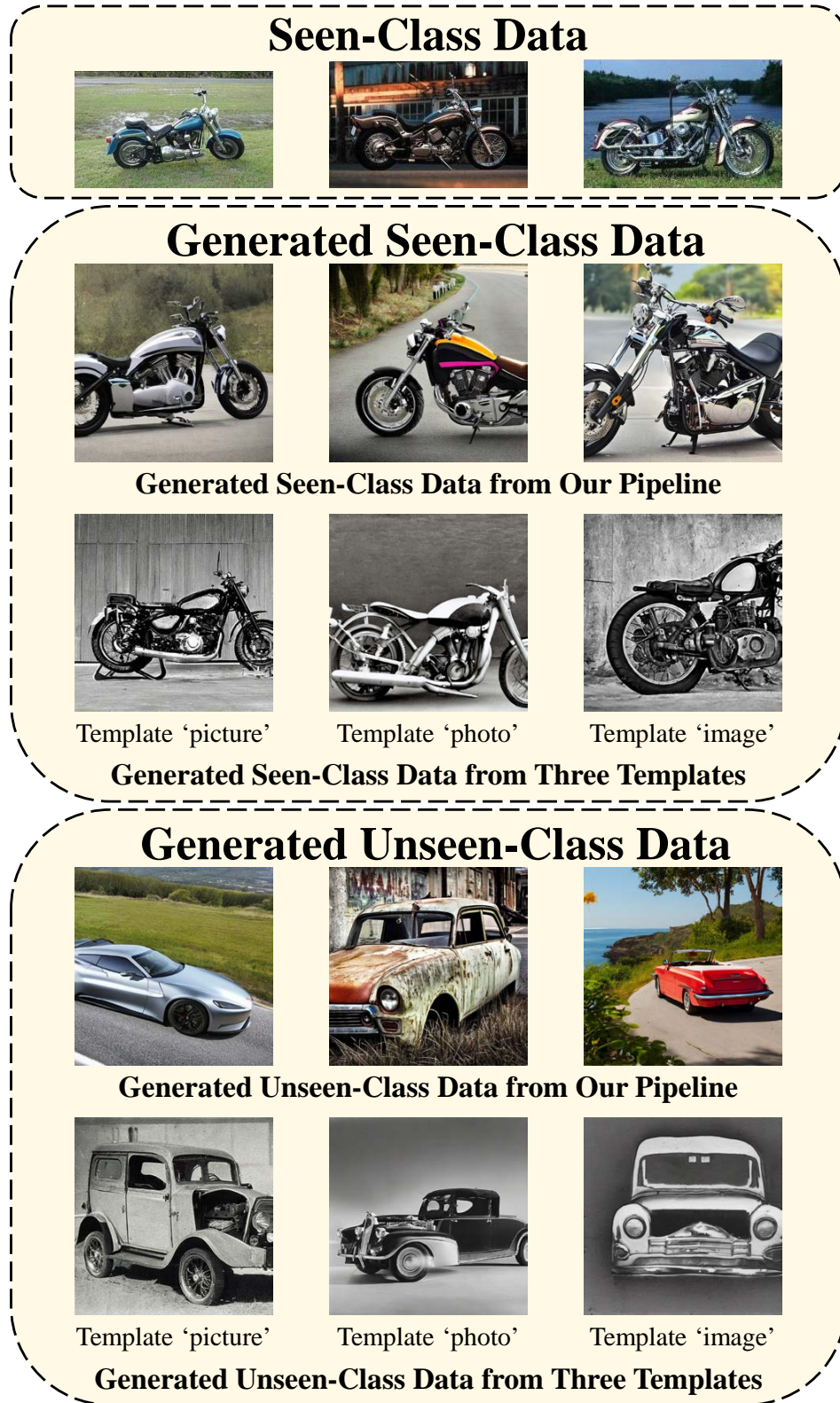


Figure 3: Comparison between the images generated with class-domain-wise data generation pipeline and three prompt templates mentioned in ablation studies. The seen class is 'motorbike' and the inferred unseen class is 'car'.

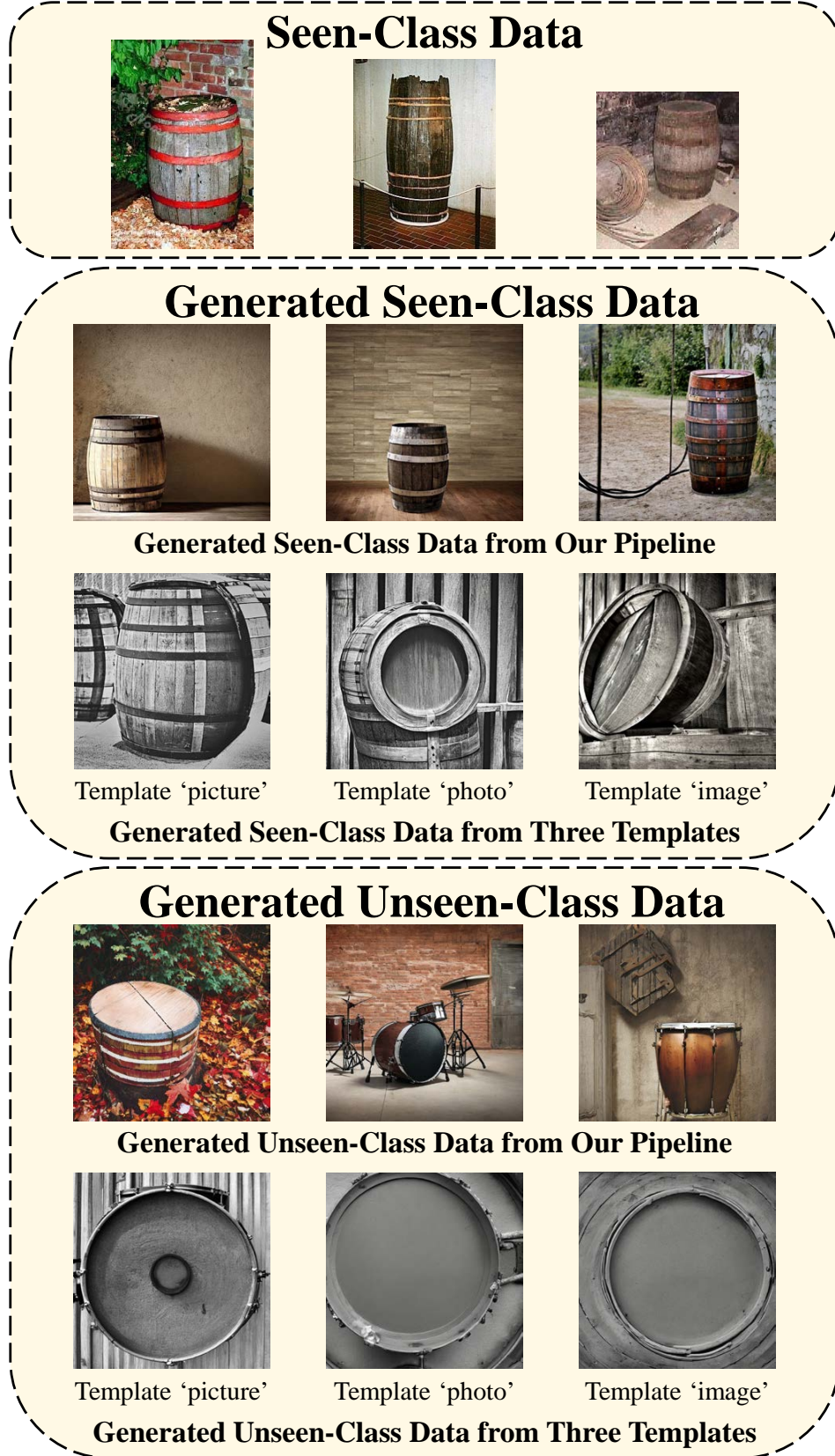


Figure 4: Comparison between the images generated with class-domain-wise data generation pipeline and three prompt templates mentioned in ablation studies. The seen class is 'barrel' and the inferred unseen class is 'drum'.

## References

- [1] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International conference on machine learning*, pages 1613–1622. PMLR, 2015.
- [2] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International conference on learning representations*, 2017.
- [3] Diederik P Kingma, Max Welling, et al. Auto-encoding variational bayes, 2013.
- [4] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in Neural Information Processing Systems*, 36:34892–34916, 2023.
- [5] David A McAllester. Some pac-bayesian theorems. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 230–234, 1998.
- [6] David A McAllester. Pac-bayesian model averaging. In *Proceedings of the twelfth annual conference on Computational learning theory*, pages 164–170, 1999.
- [7] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.
- [8] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.