A. Derivation of Theorems in Section 4.1

A.1. Theorem 4.2

Notation Statement in the Appendix. To involve risk between hypotheses and between hypothesis and ground truth models, we use $\epsilon_\cdot(h, f)$ or $\epsilon_\cdot(h, h^*)$ to specify which space the risk is computed on.

Definition A.1 ($\mathcal{H}\Delta\mathcal{H}$-distance [6])**.** *Given two feature distributions $\mathcal{D}_g$ and $\mathcal{D}_r$, and the hypothesis class $\mathcal{H}$, the $\mathcal{H}\Delta\mathcal{H}$-distance between $\mathcal{D}_g$ and $\mathcal{D}_r$ is defined as*

$$d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_g, \mathcal{D}_r) = 2 \sup_{h,h' \in \mathcal{H}} |\mathbb{P}_{\boldsymbol{x}\sim\mathcal{D}_g}[h(\boldsymbol{x}) \neq h'(\boldsymbol{x})] - \mathbb{P}_{\boldsymbol{x}\sim\mathcal{D}_r} h(\boldsymbol{x}) \neq h'(\boldsymbol{x})| \ . \tag{17}$$

We first define the expected version of Definition 4.1

Definition A.2 (Expected Generative distance)**.** *Given two feature distributions $\mathcal{D}_h$ and $\mathcal{D}_u$, the ground truth labelling function f_h, f_u, and the optimal hypothesis $h^* = \arg\min_{h\in H} \epsilon(h, f_h) + \epsilon(h, f_s)$ of a model training on the distribution $\mathcal{D}_s, \mathcal{D}_h$. The $h^*\Delta f$-distance between $\mathcal{D}_h$ and $\mathcal{D}_u$ is defined as*

$$\begin{aligned} d_{h^*}(\mathcal{D}_h, \mathcal{D}_u) = &|\mathbb{P}_{(\boldsymbol{x},\boldsymbol{a})\sim\mathcal{D}_h}[f_h(\boldsymbol{x}) \neq h^*(\boldsymbol{x}, \boldsymbol{a})] \\ &- \mathbb{P}_{(\boldsymbol{x},\boldsymbol{a})\sim\mathcal{D}_u}[f_u(\boldsymbol{x}) \neq h^*(\boldsymbol{x}, \boldsymbol{a})]| \ , \end{aligned} \tag{18}$$

Note that the following inequality related to $d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_g, \mathcal{D}_r)$ holds for any h and h^*

$$\begin{aligned} d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_g, \mathcal{D}_r) &= 2 \sup_{h,h' \in \mathcal{H}} |\mathbb{P}_{\boldsymbol{x}\sim\mathcal{D}_g}[h(\boldsymbol{x}) \neq h'(\boldsymbol{x})] - \mathbb{P}_{\boldsymbol{x}\sim\mathcal{D}_r} h(\boldsymbol{x}) \neq h'(\boldsymbol{x})| \\ &\geq 2|\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}_g}[\mathbb{1}_{h(\boldsymbol{x})\neq h^*(\boldsymbol{x})}] - \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}_r}[\mathbb{1}_{h(\boldsymbol{x})\neq h^*(\boldsymbol{x})}]| \\ &= 2|\epsilon_g(h, h^*) - \epsilon_r(h, h^*)| \ . \end{aligned} \tag{19}$$

Lemma A.3 ([1])**.** *For a fixed hypothesis, the actual risk can be estimated from the empirical error with probability $1 - \delta$*

$$\epsilon(h, f) \leq \hat{\epsilon}(h, f) + \sqrt{\frac{1}{2m} \log \frac{2}{\delta}} \ , \tag{20}$$

where $\epsilon(h, f)$ is the actual risk, $\hat{\epsilon}(h, f)$ is the empirical risk, and m is the number of testing samples.

Proposition A.4 (Bound $d_{h^*}(\mathcal{D}_u, \mathcal{D}_h)$ by $\bar{d}_{GDB}(\boldsymbol{D}_u, \boldsymbol{D}_h)$)**.** *The distribution distance $d_{h^*}(\mathcal{D}_u, \mathcal{D}_h)$ can be bounded by it empirical counterpart by*

$$d_{h^*}(\mathcal{D}_u, \mathcal{D}_h) \leq \bar{d}_{GDB}(\boldsymbol{D}_u, \boldsymbol{D}_h) + C(\frac{1}{m}, \frac{1}{\delta}) \ , \tag{21}$$

where $C(\frac{1}{m}, \frac{1}{\delta})$ is a constant term depending on the training sample size m and confidence $1 - \delta$. Here $\mathcal{D}_\cdot$ represent the distribution, and $\boldsymbol{D}_\cdot$ represents the dataset sampled from the corresponding distribution.

Proof. Similar to Equation 19, we can write our generative distance as

$$d_{h^*}(\mathcal{D}_u, \mathcal{D}_h) = 2|\epsilon_h(h^*, f) - \epsilon_u(h^*, f)| \ . \tag{22}$$

Combining Lemma A.3, we have

$$\begin{aligned} \frac{1}{2} d_{h^*}(\mathcal{D}_u, \mathcal{D}_h) &= |\epsilon_h(h^*, f) - \epsilon_u(h^*, f)| \\ &\leq |\hat{\epsilon}_h(h^*, f) - \hat{\epsilon}_u(h^*, f)| + |(\hat{\epsilon}_h(h^*, f) + \hat{\epsilon}_u(h^*, f)) - (\epsilon_h(h^*, f) + \epsilon_u(h^*, f))| \\ &\lesssim \frac{1}{2} \bar{d}_{GDB}(\boldsymbol{D}_u, \boldsymbol{D}_h) + C(\frac{1}{m}, \frac{1}{\delta}) \ , \end{aligned} \tag{23}$$

where $h^* = \arg\min_{h'\in H} \epsilon_s(h', f_s) + \epsilon_h(h', f_h)$, and $\hat{h}^* = \arg\min_{h'\in H} \hat{\epsilon}_s(h', f_s) + \hat{\epsilon}_h(h', f_h)$. Following the discussion of [8], we assume the optimal hypothesis $\hat{h}^*$ we can achieve is very close to the global minimum when the training sample is large, then we can estimate h^* in Equation 23 by $\hat{h}^*$. $C(\frac{1}{m}, \frac{1}{\delta})$ is obtained from Lemma A.3 □

Proof of theorem 4.2 Given the CZSL procedure described in section 3.1, with confidence $1-\delta$ the risk on the unseen distribution is bounded by

$$\epsilon(h, f_u^t) \leq \hat{\epsilon}(\hat{h}^*, f_s^{1:t}) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_s^{1:t}, \mathcal{D}_u^t) + \bar{\lambda} + \frac{1}{2} \bar{d}_{GDB}(\boldsymbol{D}_u^t, \boldsymbol{D}_h^t) \tag{24}$$

where $\hat{h}^* = \arg\min_{h \in H} \sum_{i=1}^t \hat{\epsilon}(h, f_s^i) + \hat{\epsilon}(h, f_h^t)$, $\bar{\lambda} = \hat{\epsilon}(\hat{h}^*, f_s^{1:t}) + \hat{\epsilon}(\hat{h}^*, f_h^t)$.

Proof. Let $h^* = \arg\min_{h \in H} \sum_{i=1}^t \epsilon(h, f_s^i) + \epsilon(h, f_h^t)$, and $\lambda = \epsilon(h^*, f_s^{1:t}) + \epsilon(h^*, f_h^t)$. We write $\epsilon_s(\cdot, \cdot)$ as the union seen distribution from time $1:t$. Then

$$\begin{aligned}
&\epsilon_u(h, f_u) \\
&= \epsilon_s(h, f_s) + \epsilon_u(h, h^*) - \epsilon_s(h, h^*) + \epsilon_h(h^*, f_h) + \epsilon_s(h^*, f_s) - \epsilon_h(h^*, f) + \epsilon_u(h^*, f) \\
&- \epsilon_s(h, f_s) - \epsilon_u(h, h^*) + \epsilon_s(h, h^*) - \epsilon_s(h^*, f_s) - \epsilon_u(h^*, f) + \epsilon_u(h, f_u) \\
&\leq \epsilon_s(h, f_s) + |\epsilon_u(h, h^*) - \epsilon_s(h, h^*)| + |\epsilon_h(h^*, f_h) + \epsilon_s(h^*, f_s)| + |\epsilon_h(h^*, f) - \epsilon_u(h^*, f)| \\
&- \epsilon_s(h, f_s) + \epsilon_s(h, h^*) - \epsilon_s(h^*, f_s) - \epsilon_u(h, h^*) + \epsilon_u(h, f_u) - \epsilon_u(h^*, f) \\
&\leq \epsilon_s(h, f_s) + d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_s^{1:t}, \mathcal{D}_u^t) + \lambda + d_{h^*}(\mathcal{D}_h, \mathcal{D}_u) - \epsilon_s(h, f_s) + \epsilon_s(h, h^*) \\
&- \epsilon_s(h^*, f_s) - \epsilon_u(h, h^*) + \epsilon_u(h, f_u) - \epsilon_u(h^*, f) \ .
\end{aligned} \tag{25}$$

Note that for any distribution

$$\begin{aligned}
|\epsilon_{\mathcal{D}}(h, f_{\mathcal{D}}) - \epsilon_{\mathcal{D}}(h, h^*)| &= |\mathbb{E}_{\boldsymbol{x} \sim \mathcal{D}}[\mathbb{1}_{h \neq f_{\mathcal{D}}}] - \mathbb{E}_{\boldsymbol{x} \sim \mathcal{D}}[\mathbb{1}_{h \neq h^*}]| \\
&= |\mathbb{E}_{\boldsymbol{x} \sim \mathcal{D}}[\mathbb{1}_{h \neq f_{\mathcal{D}}} - \mathbb{1}_{h \neq h^*}]| \\
&\leq \mathbb{E}_{\boldsymbol{x} \sim \mathcal{D}}[\mathbb{1}_{h^* \neq f_{\mathcal{D}}}] = \epsilon_{\mathcal{D}}(h^*, f_{\mathcal{D}}) \ ,
\end{aligned} \tag{26}$$

where the inequality holds by the triangle inequality of the characteristic function, *i.e.*, $\mathbb{1}[a \neq b] \geq \mathbb{1}[a \neq c] - \mathbb{1}[b \neq c]$ for $\forall a, b, c \in \mathbb{R}$. Equation (26) shows that the fourth line in Equation (25) is less than or equal to zero.

Combining Equation 26, the Equation 25 can be written as

$$\epsilon_u(h, f_u) \leq \epsilon_s(h, f_s) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_s^{1:t}, \mathcal{D}_u^t) + \lambda + d_{h^*}(\mathcal{D}_h, \mathcal{D}_u) \ , \tag{27}$$

However, Equation (27) involves unknown risk and unsolvable distribution. We combine the expected risk and the actual observed risk by Lemma A.3. Let $h^* = \arg\min_{h \in H} \sum_{i=1}^t \hat{\epsilon}(h, f_s^i) + \hat{\epsilon}(h, f_h^t)$ be the optimal hypothesis on the training set, and $\bar{\lambda} = \hat{\epsilon}(\hat{h}^*, f_s^{1:t}) + \hat{\epsilon}(\hat{h}^*, f_h^t)$, we have $\lambda \leq \bar{\lambda}$. Together with Lemma A.3 and Proposition A.4, we have

$$\epsilon_u \leq \sum_{i=1}^t \alpha^i (\hat{\epsilon}(h, f_{s^i}) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_{s^i}, \mathcal{D}_u)) + \bar{\lambda} + \frac{1}{2} \bar{d}_{GDB}(\boldsymbol{D}_u, \boldsymbol{D}_h) + C(\frac{1}{m} + \frac{1}{\delta}) \ , \tag{28}$$

□

A.2. Explanation of Statement 4.3

Let $\boldsymbol{D}_h \sim \mathcal{D}_h$ be the generated unseen set we are training on, where $\mathcal{D}_h$ is the empirical distribution of all possible generations. In unsupervised domain adaptation, [56] uses random walk to select label set for the samples who have small generalization error. Proposition 3.2 of [56] demonstrates that the self transition probability of a Markov chain represents an upper bound on the margin linear classifier's generalization error. This concept is adapted to connect our GDB bound connected to the Markov Chain in below. In our sample generation procedure, we generate only one sample from each class. Our discussion of this section will be based on this. We have $\bar{d}_{GDB}(\boldsymbol{D}_u, \boldsymbol{D}_h) \propto -\sum_{i \in I_u} \mathbb{P}(\boldsymbol{a}_{u[i]} \in \boldsymbol{D}_h)$, where the probability is taken over $\mathcal{D}_h$, and I_u is the index set of unseen real attributes. This is because the difference of the risk will be reduced if the generations contain as many points close to ground-truth unseen ones as possible. Consider the Markov chain with single step transition probabilities p_{ij} of jumping from node i to node j. Each node represents a generated sample. Let

$$p_{ij} = \mathbb{P}[h(\boldsymbol{x}_i) = y_j] \ , \tag{29}$$

where h is the hypothesis trained on $\boldsymbol{D}_h$, and the h output predictions on the current generation's classification space depending on the quality of h, and the probability is taken over $\mathcal{D}_h$. We assume the training achieves error ϵ, then $h(\boldsymbol{x}_i) = y_i$ with

probability $(1-\delta)$ if the training set contains class with attribute $\boldsymbol{a}_i$. It is not hard to prove that $\mathbb{P}(\boldsymbol{a}_{u[i]} \in \boldsymbol{D}_h) \geq p_{ii}(1-\delta)(1-\epsilon)$ by the generalization bound, since if $\boldsymbol{a}_{u[i]} \notin \boldsymbol{D}_h$, $y_{u[i]}$ is not in the current generation's classification space. It follows that

$$\bar{d}_{GDB}(\boldsymbol{D}_u, \boldsymbol{D}_h) \propto -\sum_i \mathbb{P}(\boldsymbol{a}_{u[i]} \subseteq \boldsymbol{D}_h) \leq -\sum_i p_{ii}(1-\delta)(1-\epsilon) \tag{30}$$

Then we can release the bound $\bar{d}_{GDB}(\boldsymbol{D}_u, \boldsymbol{D}_h)$ by increasing $\sum_i p_{ii}$. Note that $\mathbb{P}(\boldsymbol{a}_{u[i]} \in \boldsymbol{D}_h)$ can be replaced by $\mathbb{P}(\min_{\boldsymbol{a}_{h[j]} \in \boldsymbol{D}_h} |\boldsymbol{a}_{u[i]} - \boldsymbol{a}_{h[j]}| < \varepsilon)$ with the robustness assumption of the model.

When two generations have the same $\sum_i p_{ii}$, we prefer the one having higher diversity. The diversity of the generated set $\boldsymbol{D}_h$ can be quantified from the perspective of determinantal point process. As mentioned in [31] and [14], Determinantal Point Process (DPP) is a framework for representing a probability distribution that models diversity. More specifically, a DPP over the set $\mathcal{V}$ with $|\mathcal{V}| = N$, given a positive-definite similarity matrix $L \in \mathbb{R}^{N \times N}$, is a probability distribution P_L over any $S \subseteq \mathcal{V}$ in the following form

$$P_L[S] \propto \det(L_s) \ , \tag{31}$$

where L_s is the similarity kernel of the subset S[2]. Since the point process according to this probability distribution naturally capture the notion of diversity, we hope to generate a subset with high $P_L[\boldsymbol{D}_h]$ where the $\mathcal{V}$ is viewed as $\mathcal{D}_h$ and the transition matrix is viewed as the similarity kernel. One way to generate a set of unseen samples with high $\det(L_{\boldsymbol{D}_h})$ is to encourage the diagonality of the transition matrix, which can be achieved by promoting orthogonality of the generated samples. Moreover, since actually f_h is a look-up table, low $\sum_{j \neq i} p_{ji}$ can be explained as the large dis-similarity of the generated unseen samples from different class.

B. More Details of Section 5

Algorithm 1 shows the overall training process. The Discriminator and Generator are alternatively optimized. During the training of the Generator (line 11 – 22), we propose to generate unseen attributes (line 12 for interpolation-based method and line 12,13 for dictionary-based method) and encourage the generations to be realistic and deviate from the seen generations (line 19). After the training of each task, we propose to store the current semantic information and real features in the buffer.

B.1. Regularization terms in Loss Function 4

We closely follow [35] for the regularization terms of the Generator and Discriminator. The regularization term on discriminator encourages the semantic embedding to be close to the class center, i.e., at task t

$$\mathcal{R}_D^t = \|D(\boldsymbol{A}_s^{1:t}) - \boldsymbol{C}_s^{1:t}\|_{\mathsf{F}}^2 \ , \tag{32}$$

where $\boldsymbol{A}_s^{1:t}$ is the attribute matrix and $\boldsymbol{C}_s^{1:t}$ is the class mean matrix computed by seen features up to the current task. $\|\cdot\|_{\mathsf{F}}$ is the Frobenius norm. The regularization terms on the generator encourage the seen generations to be close to the seen class centers and have moderately distanced to their semantic neighborhoods. $\mathcal{R}_G$ is defined as

$$\mathcal{R}_G = L_{\text{nuclear}} + L_{\text{sal}} \ . \tag{33}$$

L_{nuclear} is the Nuclear loss, defined as

$$L_{\text{nuclear}} = \|\boldsymbol{C}_s^t - \boldsymbol{C}_{sg}^t\|_{\mathsf{F}}^2 \ , \tag{34}$$

where $\boldsymbol{C}_s^t$ is the class mean matrix computed by seen features of current task, and $\boldsymbol{C}_{sg}^t$ is the class mean matrix computed by generated seen features of current task. L_{sal} is the incremental bidirectional semantic alignment loss defined as

$$\begin{aligned} L_{\text{sal}} = & \frac{1}{N_s^t} \sum_{i=1}^{N_s^t} \sum_{j \in \boldsymbol{I}_i} \| \max\{0, \langle \boldsymbol{C}_{s[j]}, \boldsymbol{C}_{sg[i]} \rangle - (\langle \boldsymbol{A}_{s[i]}^t, \boldsymbol{A}_{s[j]}^t \rangle + \varepsilon)\} \|^2 \\ & + \| \max\{0, (\langle \boldsymbol{A}_{s[i]}^t, \boldsymbol{A}_{s[j]}^t \rangle - \varepsilon) - \langle \boldsymbol{C}_{s[j]}, \boldsymbol{C}_{sg[i]} \rangle\} \|^2 \ , \end{aligned} \tag{35}$$

where N_s^t is the number of current seen classes at task t, $\boldsymbol{I}_i$ is the neighbor set of class i, ε is the margin error, $\langle \cdot, \cdot \rangle$ is the cosine similarity.

[2]The feature representation of the similarity space is typically normalized so the highest eigen value is 1, and hence the determinant (multiplication of the eigen values) is < 1

Figure 5. Attribute distribution T-SNE visualizations of AWA1 dataset in different task with interpolation method

B.2. Visualization of Attribute Distribution

In our analysis, we assume that the hallucinated attributes can effectively represent the real unseen attributes compactly. To visualize the distribution of these attributes, we employ the T-SNE embedding method. As shown in Figure 5, the plot illustrates the distribution of seen attributes, unseen attributes, and hallucinated attributes across different tasks. It is important to note that only a partial subset of the hallucinated attributes for each task is displayed in the plot, while the actual number of hallucinated attributes is equivalent to the number of training samples.

As the task progresses and the learner is exposed to more seen classes, the hallucinated attributes become more aligned with the unseen attribute. However, in areas where the distribution of unseen attributes is sparse (as indicated by the blank regions in Figure 5), the hallucinated attributes are also sparse. In such cases, the hallucinated attributes tend to describe the potential visual space, deviating from the seen attributes, and providing compact support for the unseen attributes. This aligns with our assumptions and demonstrates the efficacy of our approach.

B.3. Numerical Verification of GRW Loss

In statement 3.3, we asserted that the GRW loss can effectively reduce $\bar{d}_{GDB}$. To demonstrate the relationship between the GRW loss and the bound $\bar{d}_{GDB}$, we plotted a figure using the model at different epochs for different $\hat{h}^*$. In this figure, we used the difference between the generated hallucinated samples accuracy and the test unseen accuracy to represent $\bar{d}_{GDB} = |\hat{\epsilon}_u - \hat{\epsilon}_h|$ at a randomly selected task. As shown in Figure 6, we observed a strong positive correlation between the GRW loss and $\bar{d}_{GDB}$, particularly, when the loss decreases. This finding suggests that by minimizing the GRW loss, we can reduce the bound between the generated hallucinated space and the true unseen space.

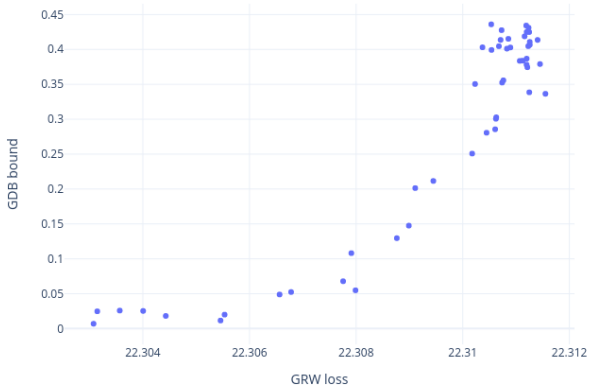

Figure 6. Relationship between $\bar{d}_{GDB}$ and the GRW loss in CUB dataset

B.4. Relation to Other Work using Random Walk

We adapt random walk modeling [5] with three key changes.

1. Previous works such as [5, 29] have represented class prototypes or centers using a few examples provided for each class. However, in our setting, we aim to deviate from seen classes and facilitate knowledge transfer to unseen classes through attributes or semantic descriptions. To achieve this, we define the seen class centers $\boldsymbol{C}$ in a semantically guided way by computing the mean of generated seen samples from their corresponding attributes. Specifically, we define $[\boldsymbol{C}]_i$ as the mean of generated samples from the attribute vector a_i for class i, i.e., $[\boldsymbol{C}]_i = \text{mean } G(z, a_i)$, where G is the generator and z is the noise vector.

2. [5, 29] use unlabeled data points to calculate the random walk, where we use generated examples.

3. In contrast to the few-shot learning problem where class prototypes are computed using unlabeled examples of seen classes, our approach generates examples from hallucinated classes. Thus, the loss functions proposed in [5, 29] aim to attract unlabeled samples to labeled samples, whereas our goal is to push hallucinated samples away from seen samples. In [29], global consistency is encouraged using a random walk from labeled data to unlabeled data (represented by their $\boldsymbol{A}$ matrix) and back to labeled samples (represented by their $\boldsymbol{A}^\top$ matrix). The aim is to promote the identity distribution of paths, where the starting and ending points are of the same class. [5] investigates a more general case where the number of random walk steps between unlabeled classes is greater than one (represented by their $\boldsymbol{B}$ matrix). In our case, as none of the generated hallucinated samples belong to seen classes, we use the random walk approach to encourage uniform distribution instead of identity distribution for all the paths from seen to generated examples of hallucinated classes and back to seen classes, represented by our $\boldsymbol{P}^{C_s X_h} \boldsymbol{P}^{X_h X_h} \boldsymbol{P}^{X_h C_s}$ matrix. This approach provides a deviation signal that encourages the model to learn distinct representations for seen and hallucinated classes, facilitating better knowledge transfer to hallucinated classes.

[56] focuses on unsupervised domain adaptation, which involves doing a random walk over all potential labeling circumstances on unlabeled target data to identify a stationary labeling distribution. Labeling stability is defined from the perspective of a generalization bound which can be attained through a stationary Markov chain. We borrow the idea of using the Markov chain to estimate the relationship between different labeling to find a stationary one that can reduce the generalization bound. We employ the Markov chain to estimate the relationship between different hallucinated generations and discover a diverse one that can reduce the generalization bound. The L_{GRW} loss encourages the random walk to find a highly diverse hallucinated generation, which in turn reduces the generalization bound.

Algorithm 1: Training procedure of ICGZSL

Input : Total task number T, training epoch E, random walk length R, decay rate of random walk γ, and coefficients $\lambda_{c,rd,i,rg}$, learning rate $\alpha_{G,D,Dic}$, buffer size B

Data : $X_s^{1:T}$, $y_s^{1:T}$, $a_s^{1:T}$

Initialize : Generator, Discriminator

for $t = 1 : T$ **do**
  Get train loader by concatenating train set t with buffer data;
  for $e = 1 : E$ **do**
    Get X_s^t, y_s^t sampled from train loader. Get $a_s^{1:t}$ from current train set and buffer ;
    begin Train Discriminator
      Generate samples conditioning on seen attributes $X_{sg}^t = G(z, a_s^t)$;
      Compute real-fake loss $\mathcal{L}_{\text{real-fake}}$ in equation (5) using real seen samples X_s^t, generated seen samples X_{sg}^t, and current task attribute a_s^t;
      Compute classification loss $\mathcal{L}_{\text{classification}}$ in equation 6 using real seen samples X_s^t, generated seen samples X_{sg}^t, and attributes $a_s^{1:t}$;
      Compute $\mathcal{L}_D$ in equation 4 and update $\theta_D \leftarrow \theta_D - \alpha_D \nabla \mathcal{L}_D$;
    end
    begin Train Generator
      Generate a_{ug}^t by interpolation between two random a_s^t ;
      if *Use dictionary based method* **then**
        Initialize the dictionary with the interpolated attribute and get θ_{Dic}
      end
      Generate samples conditioning on unseen attributes $X_{ug}^t = G(z, a_{ug}^t)$;
      Compute the second part of real-fake loss $\mathcal{L}_{\text{real-fake}}$ in equation (5) using generated unseen samples X_{sg}^t and current task attribute a_s^t;
      Compute the second part of classification loss $\mathcal{L}_{\text{classification}}$ in equation 6 using generated unseen samples X_{sg}^t and attributes $a_s^{1:t}$;
      Compute tehe inductive loss in $\mathcal{L}_{\text{inductive}}$ using $C_s^t = \text{mean}(X_s^t)$, generated seen samples X_{sg}^t, and unseen generated samples X_{ug}^t Compute $\mathcal{L}_G$ in equation 4 and update $\theta_G \leftarrow \theta_G - \alpha_G \nabla \mathcal{L}_G$;
      if *Use dictionary based method* **then**
        $\theta_{Dic} \leftarrow \theta_{Dic} - \alpha_{Dic} \nabla \mathcal{L}_D$
      end
    end
  end
  begin Replay data
    Save a_s^t to the buffer;
    Save current real features with size $B/N_s^{1:t}$ per class, reduce previous features to size $B/N_s^{1:t}$
  end
end

C. Zero-shot learning experiments

C.1. Text based zero-shot learning experiments

Text-based ZSL is more challenging because the descriptions are at the class level and are extracted from Wikipedia, which is noisier.

Benchmarks: To evaluate the efficacy of zero-shot learning (ZSL) with text descriptions as semantic class descriptions, we conducted experiments on two well-known benchmarks, namely Caltech UCSD Birds-2011 (CUB)[58] and North America Birds (NAB)[55]. While CUB contains 200 classes with 11,788 images, NAB has 1011 classes with 48,562 images. To gauge the generalization capability of class-level text zero-shot recognition, we split the benchmarks into four subsets: CUB

Metric	Seen-Unseen AUC (%)			
Dataset	CUB		NAB	
Split-Mode	Easy	Hard	Easy	Hard
ZSLNS [46]	14.7	4.4	9.3	2.3
SynC$_{fast}$ [9]	13.1	4.0	2.7	3.5
ZSLPP [18]	30.4	6.1	12.6	3.5
FeatGen [62]	34.1	7.4	21.3	5.6
LsrGAN (*tr*) [57]	39.5	12.1	23.2	6.4
+**GRW**	39.9$^{+0.4}$	13.3$^{+1.2}$	24.5$^{+1.3}$	6.7$^{+0.3}$
GAZSL (*in*) [68]	35.4	8.7	20.4	5.8
+ CIZSL [16]	39.2	11.9	24.5	6.4
+ **GRW**	**40.7** $^{+5.3}$	**13.7**$^{+5.0}$	**25.8**$^{+5.4}$	**7.4** $^{+1.6}$

Table 6. Showing Seen-Unseen AUC results of ZSL experiments on noisy text description-based datasets **CUB** and **NAB**(Easy and Hard Splits)

Setting	CUB-Easy		CUB-Hard	
	Top-1 Acc	SU-AUC	Top1-Acc	SU-AUC
+ GRW (R=1)	45.41	39.62	13.79	12.58
+ GRW (R=3)	45.11	39.25	14.21	13.22
+ GRW (R=5)	45.40	40.51	14.00	13.07
+ GRW (R=10)	**45.43**	**40.68**	**15.51**	**13.70**

Table 7. Ablation studies on CUB Dataset (text). Each row shows either baseline deviation losses and GRW losses with different length on GAZSL [68]

Metric	Top-1 Accuracy (%)				Seen-Unseen AUC (%)			
Dataset	CUB		NAB		CUB		NAB	
Split-Mode	Easy	Hard	Easy	Hard	Easy	Hard	Easy	Hard
GAZSL [68]	43.7	10.3	35.6	8.6	35.4	8.7	20.4	5.8
GAZSL [68] + GRW	**45.4**	**15.5**	**38.4**	10.1	**40.7**	**13.7**	**25.8**	**7.4**
GAZSL [68] + only L_{GRW}	45.3	14.8	38.2	**10.3**	40.1	12.8	25.8	7.4

Table 8. Ablation study using Zero-Shot recognition on **CUB** & **NAB** datasets with two split settings. We experiment with and without the $\mathcal{R}_{GRW}$ (second and last row). The first loss is the baseline method.

Easy, CUB Hard, NAB Easy, and NAB Hard. The hard splits were designed to ensure that the unseen bird classes from super-categories do not overlap with seen classes, following prior work [10, 68, 16].

Baseline and training: We introduced a novel GRW loss ($L_{GRW} + \mathcal{R}_{GRW}$) into the inductive zero-shot learning method GAZSL [68] and compared its performance with other inductive zero-shot learning methods. We employed the TF-IDF[49] representation of the input text for the text representation function $\psi(\cdot)$, followed by an FC noise suppression layer. Our experiments were conducted using a random walk length $R = 10$, and we found that longer random walk processes yield better performance in the ablation study. Each ZSL experiment was executed on a single NVIDIA P100 GPU.

Evaluation and metrics: During the test, the visual features of unseen classes are synthesized by the generator conditioned on a given unseen text description a_u, i.e. $x_u = G(s_u, z)$. We generate 60 different synthetic unseen visual features for each unseen class and apply a simple nearest neighbor classifier on top of them. We use two metrics: standard zero-shot recognition with the Top-1 unseen class accuracy and Seen-Unseen Generalized Zero-shot performance with Area under Seen-Unseen curve [10].

Results: Our proposed approach improves over older methods on all datasets and achieves SOTA on both Easy and SCE(hard) splits, as shown in Table 6. We show improvements in 0.8-1.8% Top-1 accuracy and 1-1.8% in AUC. GAZSL [68] + GRW also has an improvement of around 2% over other inductive loss (GAZSL [68] + CIZSL [16]).

GRW Loss for Transductive ZSL: To better understand how the GRW improves the consistency of generated seen features space and generated unseen features space, we conduct experiments on semantic transductive zero-shot learning settings. The improvements are solely from the GRW loss with the ground truth semantic information. We choose LsrGAN [57] as the baseline model. Our loss can also improve LsrGAN on text-based datasets on most metrics, ranging from 0.3%-3.6%. However, as we expected, the improvement in the purely inductive/more realistic setting is more significant.

Ablation: Table 7 shows the results of our ablation study on the random walk length. We find that the longer random walk performs better, giving higher accuracy and AUC scores for both easy and hard splits for CUB Dataset. With a longer random

	Top-1 Accuracy(%)			Seen-Unseen H		
	AwA2	aPY	SUN	AwA2	aPY	SUN
SJE [3]	61.9	35.2	53.7	14.4	6.9	19.8
LATEM [60]	55.8	35.2	55.3	20.0	0.2	19.5
ALE [2]	62.5	39.7	58.1	23.9	8.7	26.3
SYNC [9]	46.6	23.9	56.3	18.0	13.3	13.4
SAE [34]	54.1	8.3	40.3	2.2	0.9	11.8
DEM [67]	67.1	35.0	61.9	25.1	19.4	25.6
FeatGen [62]	54.3	42.6	60.8	17.6	21.4	24.9
cycle-(U)WGAN [20]	56.2	44.6	60.3	19.2	23.6	24.4
LsrGAN (*tr*) [57]	60.1	34.6	62.5	48.7	31.5	44.8
+ **GRW**	63.7$^{+3.6}$	35.5$^{+0.9}$	**64.2**$^{+1.7}$	**49.2**$^{+0.5}$	**32.7**$^{+1.2}$	**46.1**$^{+1.3}$
GAZSL [68]	58.9	41.1	61.3	15.4	24.0	26.7
+ CIZSL [16]	67.8	42.1	63.7	24.6	25.7	27.8
+ **GRW**	**68.4**$^{+9.5}$	**43.3**$^{+2.2}$	62.1$^{+0.8}$	39.0$^{+23.6}$	27.2$^{+3.2}$	27.9$^{+1.2}$

Table 9. Zero-Shot Recognition on class-level attributes of **AwA2**, **aPY** and **SUN** datasets, showing that GRW loss can improve the performance on attribute-based datasets.

	AWA1	AWA2	CUB	SUN
Total classes	50	50	200	705
Number of tasks	5	5	20	15
Initial seen classes	10	10	10	47
Covered class	10	10	10	47

Table 10. Seen and Unseen classes in different dataset

	AWA1		AWA2		CUB		SUN	
	Inter.	Dic.	Inter.	Dic.	Inter.	Dic.	Inter.	Dic.
λ_c	10	1	1	10	1	1	1	1
λ_i	0.5	2	1	5	2	2	5	1
R	3	3	3	3	5	5	5	5

Table 11. The hyperparameter for Table 1

walk process, the model could have a more holistic view of the generated visual representation that enables better deviation of unseen classes from seen classes.

GRW loss contains two parts, L_{GRW} and $\mathcal{R}_{GRW}$. Table 8 shows the results of our ablation study on the $\mathcal{R}_{GRW}$ in zero-shot learning. We perform experiments both with $\mathcal{R}_{GRW}$ and without $\mathcal{R}_{GRW}$. Training failed with NaN gradients in 5% of the times without $\mathcal{R}_{GRW}$ but 0% with $\mathcal{R}_{GRW}$; thus, it is important for the training stability.

C.2. Attribute based zero-shot learning experiments

Benchmarks: We perform these experiments on the AwA2 [36], aPY [19], and SUN [42] datasets.

Baseline, training, and evaluation: We perform experiments on the widely used GBU [61] setup, where we use class attributes as semantic descriptors. The evaluation process and training devices are the same as text-based experiments. We use seen accuracy, unseen accuracy, harmonic mean of seen and unseen accuracy, and top-1 accuracy as the evaluation metrics.

Results: In Table 9, we see that GRW outperforms all the existing methods on the seen-unseen harmonic mean for AwA2, aPY, and SUN datasets. In the case of the AwA2 dataset, it outperforms all the compared methods by a significant margin, i.e., 15.1% in harmonic mean, and is also competent with existing methods in Top-1 accuracy while improving 4.8%. GAZSL [68]+GRW has an average relative improvement over GAZSL [68]+CIZSL [16] and GAZSL [68] of 24.92% and 61.35% in harmonic mean.

D. Continual zero-shot learning experiments

D.1. Dataset and Continual Zero-Shot Learning Setup

We display the seen and unseen class conversions in each task for each dataset in the Table 10 to provide a better understanding of the specific implementation of CZSL on different datasets. Covered class means the number of unseen class converted to seen class per task.

	mSA		mUA		mHA	
	Mean	Std	Mean	Std	Mean	Std
AWA1	65.87	1.19	33.77	1.00	42.69	0.57
AWA2	70.52	0.46	34.52	0.90	44.45	0.79
CUB	42.11	0.88	22.10	0.67	27.80	0.53
SUN	36.29	0.18	21.07	0.33	26.44	0.20

Table 12. Our method in continual zero shot learning with interpolated attributes. Mean and variance calculated on three runs with different random seeds.

	mSA		mUA		mHA	
	Mean	Std	Mean	Std	Mean	Std
AWA1	66.35	0.28	32.75	0.94	41.90	0.91
AWA2	70.55	0.51	33.88	0.60	43.49	0.88
CUB	42.22	0.30	22.78	0.91	28.09	0.68
SUN	36.63	0.12	21.39	0.47	26.79	0.37

Table 13. Our method in inductive continual zero shot learning with learnable dictionary of attributes. Mean and variance calculated using three runs with different random seeds

D.2. More Ablations

Random seed: We experiment with multiple random seeds on the CUB dataset and show the averaged mH (line) and standard deviation (shadow) in Figure 7. The random seed mainly affects the generation part of GZSL learners. The generated data is used directly or indirectly to train the classifier of the unseen class. Figure 7 shows that previous models are sensitive to random seeds, but our model is not. Previous models use the generated data as replay data or directly train the classifier, while ours avoids these. Our method uses a non-parametric classifier, a similarity-based classifier. During training, we pay more attention to improving the generalization ability of our embedder (discriminator) by encouraging the consistency between the generated visual space and the true visual space. Plus, we store the real data in the buffer. These all make our model more stable. Although we only reported the results of one seed (2222) in Table 4, the figure shows that the effect of different seeds on the results is not significant.

We also report mean and standard deviation of multiple runs of our methods in each dataset in Table 12, 13. It shows that experiments on all the datasets with both attribute generation methods have relatively small variance. Although interpolation-based method has lower mean harmonic accuracy on fine-grained dataset CUB and SUN, it is shown to be more stable with less variance than dictionary-based method.

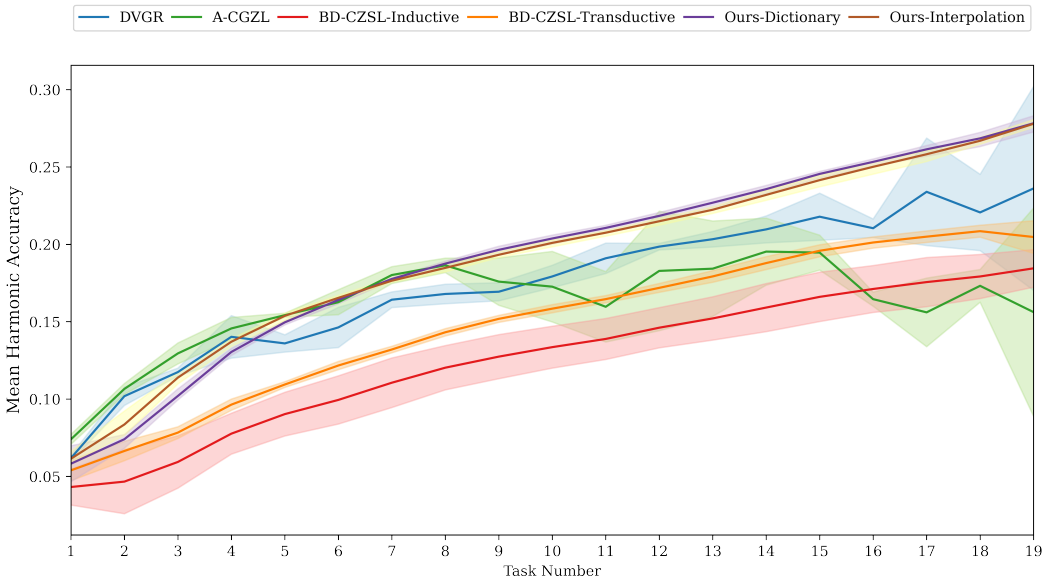

Figure 7. Mean harmonic accuracy at the end of each task with 5 different random seeds on CUB [58]. Lines show the mH, and shadows show the standard deviation.

D.3. Hyperparameters in GRW loss

Hyperparameter for Table 1: We use the validation set to tune the hyperparameter random walk step R, coefficient of $L_{\text{creativity}}$ λ_{c}, and coefficient of $L_{\text{inductive}}$ λ_{i}. The hyperparameter used to report Table 1 is shown in Table 11

Walk length R and decay rate γ: We do an ablation study on the random walk length R and decay rate γ of the GRW loss in continual zero-shot learning experiments. Table 14 shows our method with different random walk lengths in AWA1 dataset

		Ours Interpolation			Ours Dictionary		
		mSA	mUA	mHA	mSA	mUA	mHA
R	1	41.7	21.2	27.1	43.6	22.7	28.4
	3	42.3	22.1	27.7	42.1	20.6	26.6
	5	42.2	22.7	**28.4**	42.4	23.6	**28.8**

		Ours Interpolation			Ours Dictionary		
		mSA	mUA	mHA	mSA	mUA	mHA
R	1	65.4	33.9	42.7	66.6	32.7	41.5
	3	67.0	34.2	**43.4**	67.1	33.5	**42.8**
	5	65.8	33.0	42.1	66.8	32.7	41.7

Table 14. Our method with different random walk length R in AWA1 dataset (right) and CUB dataset (left)

		Ours-interpolation			Ours-dictionary		
		mSA	mUA	mH	mSA	mUA	mH
γ	0.7	40.97	21.78	27.26	42.22	22.03	27.47
	1	40.95	21.21	27.05	42.62	21.6	27.43

		Ours-interpolation			Ours-dictionary		
		mSA	mUA	mH	mSA	mUA	mH
γ	0.7	66.8	33.42	42.87	66.93	32.41	41.51
	1	66.07	32.31	41.69	66.34	32.87	41.76

Table 15. Our method with different decay rate γ on CUB dataset (left) and AWA1 dataset (right)

		Ours-interpolation			Ours-dictionary		
		mSA	mUA	mH	mSA	mUA	mH
λ_i	0.01	41.81	20.93	27.01	42.8	23.07	28.51
	0.1	42.32	21.27	27.11	42.73	21.98	27.85
	1	40.97	21.78	27.26	42.22	22.03	27.47

		Ours-interpolation			Ours-dictionary		
		mSA	mUA	mH	mSA	mUA	mH
λ_i	0.1	66.81	32.82	42.15	66.32	32.11	41.15
	1	66.8	33.42	42.87	66.93	32.41	41.51
	10	66.38	33.77	42.92	66.47	31.81	40.89

Table 16. Our method with different inductive coefficients λ_i on CUB dataset (left) and AWA1 dataset (right)

and CUB dataset. In the dataset AWA1, moderate lengths give the highest mHA while in the CUB dataset higher random walk lengths provide the best mHA. It shows that the more challenging the dataset, the more random walk length is needed. Unlike ZSL experiments, in CZSL experiments, knowledge is not only transferred to the unseen class space but also to the next task. Long walk length could give the model a more holistic view of the current task, but may harm the transformation to the next task. Therefore, tuning the number of random walk steps is required for new datasets.

Decay rate γ works as a scale factor to the GRW loss to prevent a specific area in the probability matrix from being too close to one, resulting in exponential growth in the multiplication results when compared to other areas. Compared to the non-decay case when $\gamma = 1$ in Table 15, the decayed case has noticeable improvements in unseen accuracy, resulting in better harmonic accuracy.

D.4. Ablations on Weight of Inductive Loss

Inductive weight λ_i We also do an ablation study on the inductive coefficient λ_i in Table 16. This factor mainly affects the proportion of inductive loss in the overall loss. We found that our model is not sensitive to this hyperparameter. Whether on the larger dataset CUB or the smaller dataset AWA1, the difference of mH of different λ_i on our model does not exceed 1%. Therefore, our model does not need too much parameter tuning process.

D.5. Continual zero-shot learning with other common settings

Although our main research problem is inductive setting, and we think real replay is needed, we still have an open attitude to other settings and migrate our model naively to their setting. We show experiment results in these settings in Table 17 and compare them with other methods.

We mentioned earlier that the generative replay method has unbalanced storage and buffer overload problems, but many models still use generative replay. When data privacy concerns are encountered, the generative replay method may be an alternative to the real replay method. When using the generative replay, our model outperforms most existing methods. Our problem analysis cannot be applied in this setting, since we believe the replayed feature should have a balanced number in each class.

Our primary focus is on the inductive setting, but we also provide results in the transductive setting and with generative replay. In the transductive setting, we use the ground truth unseen attributes to generate the visual features, and our loss works on these generations. Our method is comparable with other transductive methods, even without carefully designing how to use the semantic information.

Through these knots, we believe that our model has the possibility of being migrated to other settings and is valuable for further explorations in other settings.

Table 17. Comparison of our inductive loss in other common CZSL settings

	replay method	zsl setting	AWA1			AWA2			CUB			SUN		
			mSA	mUA	mHA	mSA	mUA	mHA	mSA	mUA	mHA	mSA	mUA	mHA
CN-ZSL	real	in	-	-	-	33.55	6.44	10.77	44.31	14.8	22.7	22.18	8.24	12.46
Ours-interpolation	real	in	62.9	32.77	**42.03**	67.41	35.4	**45.06**	40.17	21.78	27.26	36.29	21.05	26.51
Ours-dictionary	real	in	63.43	32	41.15	68.02	33.22	42.89	41.45	22.03	**27.47**	36.54	21.31	**26.76**
DVGR	generative	tr	65.1	28.5	38	73.5	28.8	40.6	44.87	14.55	21.66	22.36	10.67	14.54
A-CGZSL	generative	tr	70.16	25.93	37.19	70.16	25.93	37.19	34.25	12.42	17.41	17.2	6.31	9.68
BD-CGZSL	generative	tr	67.55	36.04	**47.88**	71.37	38.76	**51.6**	31	23.97	**26.01**	30.08	20.07	23.72
Ours-interpolation	generative	tr	62.43	33.03	42.01	66.84	34.01	43.77	32.53	16.66	21.65	-	-	-
Ours-dictionary	generative	tr	62.34	31.5	40.18	68.07	34.45	44.17	30	16.18	20.55	-	-	-
BD-CGZSL-in	generative	in	62.12	31.51	40.46	67.68	32.88	42.33	37.76	9.089	14.43	34.93	14.86	20.8
Ours-interpolation	generative	in	61.43	34.04	**42.18**	67.34	35.29	**44.95**	29.78	16.86	**21.06**	30.9	18.4	**22.99**
Ours-dictionary	generative	in	62.26	30.88	39.68	67.44	33.68	43.24	28.34	16.94	20.57	30.13	18.56	22.85