Learn to Categorize or Categorize to Learn? Self-Coding for Generalized Category Discovery
Supplementary Materials

Sarah Rastegar, Hazel Doughty[*], Cees G. M. Snoek
University of Amsterdam

Contents

[*]Currently at Leiden University

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

1 Theory

1.1 Notation and Definitions

Let us first formalize our notation and definition for the rest of the section. Some definitions might overlap with the notations in the main paper. However, we repeat them here for ease of access.

Probabilistic Notations. We denote the input random variable with X and the category random variable with C. The category code random variable, which we define as the embedding sequence of input X^i, is denoted by $\mathrm{z}^i = z_1^i z_2^i \cdots z_L^i$, in which superscript i shows the ith sample, while subscript L shows the digit position in the code sequence.

Coding Notations. Let $\mathcal{C}$ be a countable set, we use $\mathcal{C}^*$ to show all possible finite sequences using the members of this set. For instance: $\{0,1\}^* = \{\epsilon, 0, 1, 00, 01, 10, 11, \cdots\}$ in which ϵ is empty word. The length of each sequence z, which we show with $l(\mathrm{z})$, equals the number of digits present in that sequence. For instance, for the sequence $l(01010) = 5$.

Shannon Information Theory Notations. We denote the *Shannon entropy* or *entropy* of the random variable X with $H(X)$. It measures the randomness of values of X when we only have knowledge about its distribution P. It also measures the minimum number of bits required on average to transmit or encode the values drawn from this probability distribution [1, 2]. The *conditional entropy* of a random variable X given random variable Z is shown by $H(X|Z)$, which states the amount of randomness we expect to see from X after observing Z. In addition, $I(X;Z)$ indicates the *mutual information* between random variables X and Z [1, 2], which measures the amount of *information* we can obtain for one random variable by observing the other one. Note that contrary to $H(X|Z)$, mutual information is *symmetric*.

Algorithmic Information Theory Notations. Similar to Shannon's information theory, *Kolmogorov Complexity* or *Algorithmic Information Theory*[3–5] measures the shortest length to describe an object. Their difference is that Shannon's information considers that the objects can be described by the characteristic of the source that produces them, but *Kolmogorov Complexity* considers that the description of each object in isolation can be used to describe it with minimum length. For example, a binary string consisting of one thousand zeros might be assigned a code based on the underlying distribution it has been drawn from. However, *Kolmogorov Complexity* shows that we can encode this particular observation by transforming a description such as `"print 0 for 1000 times"`. The analogon to entropy is called *complexity* $K(\mathrm{x})$, which specifies the minimum length of a sequence that can *specify* output for a particular system. We denote the *algorithmic mutual information* for sequences x and z with $I_{alg}(\mathrm{x} : \mathrm{z})$, which specifies how much information about sequence x we can obtain by observing sequence z.

1.2 Maximizing the Algorithmic Mutual Information

Let's consider data space $\mathcal{D}=\{X^i, C^i : i \in \{1, \cdots N\}\}$ where Xs are inputs and Cs are the corresponding category labels.

Lemma 1 *For each category c and for X^i with C^i=c, we can find a binary decision tree $\mathcal{T}_\mathrm{c}$ that starting from its root, reaches each X^i by following the decision tree path. Based on this path, we assign code $c(X^i)$=$c_1^i c_2^i \cdots c_M^i$ to each X^i to uniquely define and retrieve it from the tree.*

Proof of Lemma 1. Since the number of examples in the dataset is finite, we can enumerate samples of category c with any arbitrary coding. We then can replace these enumerations with their binary equivalent codes. We start from a root, and every time we encounter 1 in digits of these codes, we add a right child node, and for 0, we add a left child node. We then continue from the child node until we reach the code's end. Since the number of samples with category c is limited, this process should terminate. On the other hand, since the binary codes for different samples are different, these paths are unique, and by the time we traverse a path from the root to a leaf node, we can identify the unique sample corresponding to that node. □

As mentioned in the main paper, using this Lemma, we can find at least one supertree $\mathbf{T}$ for the whole data space that addresses all samples in which samples of the same category share a similar prefix. We can define a model that provides the binary code z^i=$z_1^i \cdots z_L^i$ for data input X^i with category c based on the path it takes in these eligible trees. We define these path encoding functions *valid encoding* as defined in Definition 1:

Definition 1 *A valid encoding for input space $\mathcal{X}$ and category space $\mathcal{C}$ is defined as an encoding that uniquely identifies every $X^i \in \mathcal{X}$. At the same time, for each category $c \in \mathcal{C}$, it ensures that there is a sequence that is shared among all members of this category but no member out of the category.*

Since there is no condition on how to create these trees and their subtrees, many candidate trees can address the whole data space while preserving a similar prefix for the members of each category.

However, based on our inspirations for how the brain does categorization, we assume the ground truth underlying tree $\mathbf{T}$ has a minimum average length path from the root to each node. In other words, each sample x has the shortest description code z to describe that data point while maintaining its validity. If we use a model to learn this encoding, the optimal model tree should be isomorph to the underlying tree $\mathbf{T}$,

Lemma 2 *For a learned binary code z^i to address input X^i, uniquely, if the decision tree of this encoding is optimal, it is isomorph to the underlying tree T.*

Proof of Lemma 2. Since the underlying tree has the minimum Kolmogorov complexity for each sample, we can extract the optimal lengths of each sample by traversing the tree. Evans and Lanoue [6] showed that a tree can be recovered from the sequence of lengths of the paths from the root to leaves to the level of isomorphism. Based on our assumption about the underlying tree $\mathbf{T}$, the optimal tree can not have a shorter length for any sample codes than the underlying tree. On the other hand, having longer codes contradicts its optimality. Hence the optimal tree should have similar path lengths to the underlying ground truth tree. Therefore, it is isomorphic to the underlying tree. □

Since the optimal tree with the valid encoding $\tilde{z}$ is isomorph to the underlying tree, we will have the necessary conditions that Theorem 1 provides.

Theorem 1 *For a learned binary code z^i to address input x^i, uniquely, if the decision tree of this encoding is isomorph to underlying tree $\mathbf{T}$, we will have the following necessary conditions:*

1. $I_{alg}(\mathrm{z} : x) \geq I_{alg}(\tilde{\mathrm{z}} : x) \quad \forall \tilde{z}, \tilde{z}$ *is a valid encoding for* x
2. $I_{alg}(\mathrm{z} : c) \geq I_{alg}(\tilde{\mathrm{z}} : c) \quad \forall \tilde{z}, \tilde{z}$ *is a valid encoding for* x

Proof of Theorem 1.

Part one: From the way $\mathbf{T}$ has been constructed, we know that $K(x|\mathbf{T}) \leq K(x|\mathcal{T})$ in which $\mathcal{T}$ is an arbitrary tree. From the complexity and mutual information properties, we also have $I_{\text{alg}}(\mathrm{z} : x) = K(\mathrm{z}) - K(x|\mathrm{z})$ [7]. Since $\tilde{z}$ and z have isomorph tree structures, then $K(\tilde{z}) = K(\mathrm{z})$, hence: $I_{\text{alg}}(\mathrm{z} : x) \geq I_{\text{alg}}(\tilde{\mathrm{z}} : x)$. □

Part two: In any tree that is a valid encoding, all samples of a category should be the descendants of that node. Thus, the path length to corresponding nodes should be similar in both trees. Otherwise, the length of the path to all samples of this category will not be optimal. We can use the same logic and deduce that the subtree with the category nodes as its leaves would be isomorph for both embeddings. Let's denote the path from the root to category nodes with z_c and from the category node to its corresponding samples with z_x. If we assume these two paths can be considered independent, we will have $K(x) = K(\mathrm{z}_c\mathrm{z}_x) = K(\mathrm{z}_c) + K(\mathrm{z}_x)$, which indicates that minimizing $K(x)$ in the tree implies that $K(c)$ also should be minimized. By applying the same logic as part one, we can deduce that $I_{\text{alg}}(\mathrm{z} : c) \geq I_{\text{alg}}(\tilde{\mathrm{z}} : c)$. □

1.2.1 Shannon Mutual Information Approximation

Optimization in Theorem 1 is generally not computable [3–5, 8]. However, We can approximate these requirements using Shannon mutual information instead. Let's consider two functions f and g, such that both are $\{0,1\}^* \to \mathbb{R}$. For these functions, $f \overset{+}{<} g$ means that there exists a constant κ, such that $f \leq g + c$, when both $f \overset{+}{<} g$ and $g \overset{+}{<} f$ hold, then $f \overset{\pm}{=} g$ [7].

Theorem 2 *[7] Let P be a computable probability distribution on $\{0,1\}^* \times \{0,1\}^*$. Then:*

$$I(X;Z) - K(P) \overset{+}{<} \sum_x \sum_z p(x,z) I_{alg}(\mathrm{x} : \mathrm{z}) \overset{+}{<} I(X;Z) + 2K(P) \tag{1}$$

This theorem states that the expected value of algorithmic mutual information is close to its probabilistic counterpart. This means that if we maximize the Shannon information, we also approximately maximize the algorithmic information and vice versa.

Since Shannon entropy does not consider the inner regularity of the symbols it codes, to make each sequence meaningful from a probabilistic perspective, we convert each sequence to an equivalent random variable number by considering its binary digit representation. To this end, we consider $Z^i{=}\sum_{k=1}^{m}\frac{z_k^i}{2^k}$, which is a number between 0 and 1. Note that we can recover the sequence from the value of this random variable. Since the differences in the first bits affect the number more, for different error thresholds, Shannon's information will focus on the initial bits more. In dealing with real-world data, the first bits of encoding of a category sequence are more valuable than later ones due to the hierarchical nature of categories. Furthermore, with this tweak, we equip Shannon's model with a knowledge of different positions of digits in a sequence. To replace the first item of Theorem 1 by its equivalent Shannon mutual information, we must also ensure that z has the minimum length. For the moment, let's assume we know this length by the function $l(X^i){=}l_i$. Instead of Z^i, we can consider its truncated form $Z_{l_i}^i{=}\sum_{k=1}^{l_i}\frac{z_k^i}{2^k}$. This term, which we call the address loss function, is defined as follows:

$$\mathcal{L}_{\text{adr}} = -\frac{1}{N}\sum_{i=0}^{N} I(X^i; Z_{l_i}^i) \quad s.t. \quad Z_{l_i}^i = \sum_{k=1}^{l_i}\frac{z_k^i}{2^k} \text{ and } \forall k, z_k^i \in \{0,1\}. \tag{2}$$

We can approximate this optimization with a reconstruction or contrastive loss.

1.2.2 Approximation with Reconstruction Loss

Let's approximate the maximization of the mutual information by minimizing the $\mathcal{L}_{MSE}$ of the reconstruction from the code z. Suppose that $D(X)$ is the decoder function, and it is a Lipschitz continuous function, which is a valid assumption for most deep networks with conventional activation functions [9]. We can find an upper bound for $\mathcal{L}_{MSE}$ using Lemma 3.

Lemma 3 *Suppose that $D(X)$ is a Lipschitz continuous function with Lipschitz constant κ, then we will have the following upper bound for $\mathcal{L}_{MSE}$:*

$$\mathcal{L}_{MSE}(X) \leq \kappa\frac{1}{N}\sum_{i=0}^{N} 2^{-2l_i}$$

Proof of Lemma 3. Let's consider the $\mathcal{L}_{MSE}$ loss for the reconstruction $\hat{X}^i$ from the code Z^i. We denote reconstruction from the truncated category code $Z_{l_i}^i$ with $\hat{X}_{l_i}^i$.

$$\mathcal{L}_{MSE}(X) = \frac{1}{N}\sum_{i=0}^{N} \parallel \hat{X}_{l_i}^i - X^i \parallel^2$$

If we expand this loss, we will have the following:

$$\begin{aligned}\mathcal{L}_{MSE}(X) &= \frac{1}{N}\sum_{i=0}^{N} \parallel D(Z_{L(X^i)}^i) - X^i \parallel^2 \\ &= \frac{1}{N}\sum_{i=0}^{N} \parallel D(\sum_{k=0}^{l_i}\frac{z_k^i}{2^k}) - X^i \parallel^2\end{aligned}$$

Let's assume the optimal model can reconstruct X^i using the entire code length Z^i, i.e. $X^i = D(\sum_{k=0}^{m}\frac{z_k^i}{2^k})$. Now let's replace this in the equation:

$$\mathcal{L}_{MSE}(X) = \frac{1}{N}\sum_{i=0}^{N} \parallel D(\sum_{k=0}^{l_i}\frac{z_k^i}{2^k}) - D(\sum_{k=0}^{m}\frac{z_k^i}{2^k}) \parallel^2$$

Given that $D(X)$ is a Lipschitz continuous function with the Lipschitz constant κ, then we will have the following:

$$\begin{aligned}\mathcal{L}_{MSE}(X) \leq & \kappa \frac{1}{N} \sum_{i=0}^{N} \| \sum_{k=0}^{l_i} \frac{z_k^i}{2^k} - \sum_{k=0}^{m} \frac{z_k^i}{2^k} \|^2 \\ \leq & \kappa \frac{1}{N} \sum_{i=0}^{N} \| \, 2^{-l_i} \, \|^2 \\ = & \kappa \frac{1}{N} \sum_{i=0}^{N} 2^{-2l_i} \qquad \square\end{aligned}$$

Lemma 3 indicates that to minimize the upper bound on $\mathcal{L}_{MSE}$, we should aim for codes with maximum length, which can also be seen intuitively. The more length of latent code we preserve, the more accurate the reconstruction would be. This is in direct contrast with the length minimization of the algorithmic mutual information. So, the tradeoff between these two objectives defines the optimal final length of the category codes.

1.2.3 Approximation with Contrastive Loss

One of the advantages of contrastive learning is to find a representation that maximizes the mutual information with the input [10]. More precisely, if for input X^i, we show the hidden representation learning Z^i, that is learned contrastively by minimizing the InfoNCE loss, [10] showed that the following lower bound on mutual information exists:

$$I(X^i; Z^i) \geq \log(N) - \mathcal{L}_N. \tag{3}$$

Here, $\mathcal{L}_N$ is the InfoNCE loss, and N indicates the sample size consisting of one positive and $N-1$ negative samples. Equation 3 shows that contrastive learning with the InfoNCE loss can be a suitable choice for minimizing the $\mathcal{L}_{adr}$ in Equation 2. We will use this to our advantage on two different levels. Let's consider that Z^i has dimension d, and each latent variable z_k^i can take up n different values. The complexity of the feature space for this latent variable would be $\mathcal{O}(n^d)$, then the number of structurally different binary trees for this feature space would be $\mathcal{O}(C_{n^d})$, in which C_i is the ith Catalan number, which asymptotically grows as $\mathcal{O}(4^i)$. Hence the number of possible binary taxonomies for the categories will be $\mathcal{O}(4^{n^d})$. So minimizing n and, to a lesser degree, d, will be the most effective way to limit the number of possible binary trees. Since our model and the amount of training data is bounded, we must minimize the possible search space while still providing reasonable performance. On the other hand, the input feature space X^i with N possible values and dimension D has $\mathcal{O}(N^D)$ possible states, and to cover it completely, we can not arbitrarily decrease d and n. Note that for a nearly continuous function $N \to \infty$, the probability of a random discrete tree to fully covering this space would be near zero.

1.3 Category Code Length Minimization

In the main paper, we indicate the code length loss $\mathcal{L}_{length}$, which we define as $\mathcal{L}_{\text{length}} = \frac{1}{N} \sum_{i=0}^{N} l_i$. To minimize this loss, we define a binary mask sequence $\mathrm{m}^i = m_1^i m_2^i \cdots m_L^i$ to simulate the subscript property of l_i. We discussed minimizing the L_p Norm for the weighted version of the mask, which we denote with $\bar{\mathrm{m}}^i = (m_1^i 2^1)(m_2^i 2^2) \cdots (m_L^i 2^L)$. This will ensure the requirements because adding one extra bit has an equivalent loss of all previous bits.

$$\mathcal{L}_{\text{length}} \approx \frac{1}{N} \sum_{i=0}^{N} \| \, \bar{\mathrm{m}}^i \, \|_p \, . \tag{4}$$

Lemma 4 *Consider the weighted mask* $\bar{\mathrm{m}} = (m_1 2^1)(m_2 2^2) \cdots (m_L 2^L)$ *where* m_j*s are* 0 *or* 1*. Consider the norm* $\| \, \bar{\mathrm{m}} \, \|_p$ *where* $p \geq 1$*, the rightmost* 1 *digit contributes more to the norm than the entire left sequence.*

Proof of Lemma 4. Let's consider the loss function for mask $\bar{\mathrm{m}}=(m_1 2^1)(m_2 2^2)\cdots(m_L 2^L)$ and let's denote the rightmost 1 index, with k, for simplicity we consider the $\| \bar{\mathrm{m}} \|_p^p$:

$$\| \bar{\mathrm{m}} \|_p^p = \sum_{j=0}^{L}(m_j 2^j)^p = \sum_{j=0}^{k-1}(m_j 2^j)^p + (m_k 2^k)^p + \sum_{j=k+1}^{L}(m_j 2^j)^p$$

given that $m_j = 0, \forall j > k$ and $m_k = 1$, we will have:

$$\| \bar{\mathrm{m}} \|_p^p == \sum_{j=0}^{k-1}(m_j 2^j)^p + 2^{kp} + 0$$

now let's compare the two subparts of the right-hand side with each other:

$$\sum_{j=0}^{k-1}(m_j 2^j)^p \leq \sum_{j=0}^{k-1}(2^j)^p = \frac{2^{kp}-1}{2^p-1} < 2^{kp} \qquad \square$$

Hence $\mathcal{L}_{Length}$ tries to minimize the position of the rightmost 1, simulating the cutting length subscript.

1.3.1 Satisfying Binary Constraints.

In the main paper, we stated that we have two conditions, *Code Constraint:*$\forall z_k^i,\ z_k^i = 0$ *or* $z_k^i = 1$ and *Mask Constraint* $\forall m_k^i,\ m_k^i = 0$ *or* $m_k^i = 1$. We formulate each constraint in an equivalent Lagrangian function to make sure they are satisfied. For the binary code constraint we consider $f_{\text{code}}(z_k^i)=(z_k^i)(1 - z_k^i)=0$, which is only zero if $z_k^i=0$ or $z_k^i=1$. Similarly, for the binary mask constraint, we have $f_{\text{mask}}(m_k^i)=(m_k^i)(1-m_k^i)=0$. To ensure these constraints are satisfied, we optimize them with the Lagrangian function of the overall loss. Consider the Lagrangian function for $\mathcal{L}_{\text{total}}$,

$$\mathbf{L}(\mathcal{L}_{\text{total}}, \eta, \mu) = \mathcal{L}_{\text{total}} + \eta\mathcal{L}_{\text{code\_cond}} + \mu\mathcal{L}_{\text{mask\_cond}}$$

This lagrangian function ensures that constraints are satisfied for $\eta \to +\infty$ and $\mu \to +\infty$. Note that our method uses a tanh activation function which has been mapped between 0 and 1, to produce m_k and z_k, so the conditions are always greater or equal to zero. For an unbounded output, we can consider the squared version of constraint functions to ensure that constraints will be satisfied. This shows how we reach the final unconstrained loss function in the paper.

2 Experiments

2.1 Dataset Details

To acquire the train and test splits, we follow [11]. We subsample the training dataset in a ratio of 50% of known categories at the train and all samples of unknown categories. For all datasets except CIFAR100, we consider 50% of categories as known categories at training time. For CIFAR100 as in [11] 80% of the categories are known during training time. A summary of dataset statistics and their train test splits is shown in Table 1.

CIFAR10/100[12] are coarse-grained datasets consisting of general categories such as *car, ship, airplane, truck, horse, deer, cat, dog, frog* and *bird*.

ImageNet-100 is a subset of 100 categories from the coarse-grained ImageNet [13] dataset.

CUB or the Caltech-UCSD Birds-200-2011 (CUB-200-2011) [14] is one of the most used datasets for fine-grained image recognition. It contains different bird species, which should be distinguished by relying on subtle details.

FGVC-Aircraft or Fine-Grained Visual Classification of Aircraft [15] dataset is another fine-grained dataset, which, instead of animals, relies on airplanes. This might be challenging for image recognition models since, in this dataset, structure changes with design.

SCars or Stanford Cars [16] is a fine-grained dataset of different brands of cars. This is challenging since the same brand of cars can look different from different angles or with different colors.

Table 1: **Statistics of datasets and their data splits for the generalized category discovery task.** The first three datasets are coarse-grained image classification datasets, while the next four are fine-grained datasets. The Herbarium19 dataset is both fine-grained and long-tailed.

	Labelled		**Unlabelled**	
Dataset	#Images	#Categories	#Images	#Categories
CIFAR-10 [12]	12.5K	5	37.5K	10
CIFAR-100 [12]	20.0K	80	30.0K	100
ImageNet-100 [13]	31.9K	50	95.3K	100
CUB-200 [14]	1.5K	100	4.5K	200
SCars [16]	2.0K	98	6.1K	196
Aircraft [15]	3.4K	50	6.6K	100
Oxford-Pet [17]	0.9K	19	2.7K	37
Herbarium19 [18]	8.9K	341	25.4K	683

Oxford-Pet [17] is a fine-grained dataset of different species of cats and dogs. This is challenging since the amount of data is very limited in this dataset, which makes it prone to overfitting.

Herbarium_19 [18] is a botanical research dataset about different types of plants. Due to its long-tailed alongside fine-grained nature, it is a challenging dataset for discovering novel categories.

2.2 Implementation details

In this section, we provide our implementation details for each block separately. As mentioned in the main paper, the final loss function that we use to train the model is:

$$\mathcal{L}_{\text{final}} = \mathcal{L}_{\text{adr}} + \delta\mathcal{L}_{\text{length}} + \eta\mathcal{L}_{\text{Cat}} + \zeta\mathcal{L}_{\text{code\_cond}} + \mu\mathcal{L}_{\text{mask\_cond}}. \tag{5}$$

In which the loss $\mathcal{L}_{\text{adr}}$ is:

$$\mathcal{L}_{\text{adr}} = \alpha\mathcal{L}_{\text{C\_in}} + \beta\mathcal{L}_{\text{C\_code}}. \tag{6}$$

In this formula, $\mathcal{L}_{\text{C\_in}}$ is the loss function that [11] suggested, so we use the same hyperparameters as their defaults for this loss. Hence, we only expand on $\mathcal{L}_{\text{C\_code}}$:

$$\mathcal{L}_{\text{adr}} = \alpha\mathcal{L}_{\text{C\_in}} + \beta((1-\lambda_{\text{code}})\mathcal{L}^{u}_{\text{C\_code}} + \lambda_{\text{code}}\mathcal{L}^{s}_{\text{C\_code}}). \tag{7}$$

In the scope of our experimentation, it was assumed by default that α=1 and λ_{code}=0.35. The code generation process introduces a certain noise level, potentially leading to confusion in the model, particularly in fine-grained data. To mitigate this, we integrated a smoothing hyperparameter within our contrastive learning framework, aiming to balance the noise impact and avert excessive confidence in the generated code, for datasets such as CUB and Pet, the smoothing factor was set at 1, whereas for SCars, Aircraft, and Herb datasets, it was adjusted to 0.1. In contrast, we did not apply smoothing for generic datasets like CIFAR 10/100 and ImageNet, where label noise is less significant.

Furthermore, in dealing with fine-grained data, we opted to fine-tune the final two blocks of the DINO model. This approach differs from our strategy for generic datasets, where only the last block underwent fine-tuning. Additionally, we employed semi-supervised k-means at every epoch to derive pseudo-labels from unlabeled data. These pseudo-labels were then used in our supervised contrastive learning process as a supervisory signal. It is important to note that in supervised contrastive learning, the primary requirement is that paired samples belong to the same class, allowing us to disregard discrepancies between novel class pseudo-labels and their actual ground truth values. Furthermore, instead of cosine similarity for contrastive learning, we adopt Euclidean distance, a better approximation for the category problem. Finally, for balanced datasets, we use the balanced version of k-means for semi-supervised k-means.

Code Generator. To create this block, we use a fully connected network with GeLU activation functions [19]. Then, we apply a tanh activation function $\tanh(ax)$ in which a is a hyperparameter showing the model's age. We expect that as the model's age increases or, in other words, in later epochs, the model will be more decisive because of sharper transitions from 0 to 1. Hence, we will have a stronger binary dichotomy for code values. Also, since contrastive learning makes the different samples as far as possible, this causes a problem for the Code Generator because the feature space will not smoothly transition from different samples of the same category, especially for fine-grained datasets. To alleviate this problem, we use a label smoothing hyperparameter in the contrastive objective to help make feature space smoother, which will require a smaller tree for encoding. Since the model should distinguish 0s for the mask from 0s of the code, we do not adjust the code generator to 0 and 1s and consider the -1 and 1 values in practice.

Code Masker. The *Code Masker* block is a fully connected network with tanh activation functions at the end, which are adjusted to be 0 and 1s. We also consider the aging hyperparameter for the tanh activation function in the masking block. In the beginning, since codes are not learned, masking the embedding space might hamper its learning ability. To solve this, we start masker with all one's entries and gradually decrease it with epochs. Hence, the activation function that is applied to the masker would be $\tanh(x + \frac{1}{a+1})$, in which a is the aging parameter. In practice, we observed that norm one is stable enough in this loss function while also truncating codes at a reasonable length. Since $\mathcal{L}_{\text{length}}$ grows exponentially with code length, it will mask most of the code. For fine-grained datasets, this could be detrimental for very similar categories. To alleviate this problem, instead of using 2 as a positional base, we decrease it with each epoch to $2 - \frac{\text{epoch}}{N_{\text{epochs}}}$. So, at the end of training, the values of all positions are the same. This allows the model to encode more levels to the tree. Since we start with the base 2, we are constructing the tree with a focus on nodes near the root at the start and to the leaves at the end of training.

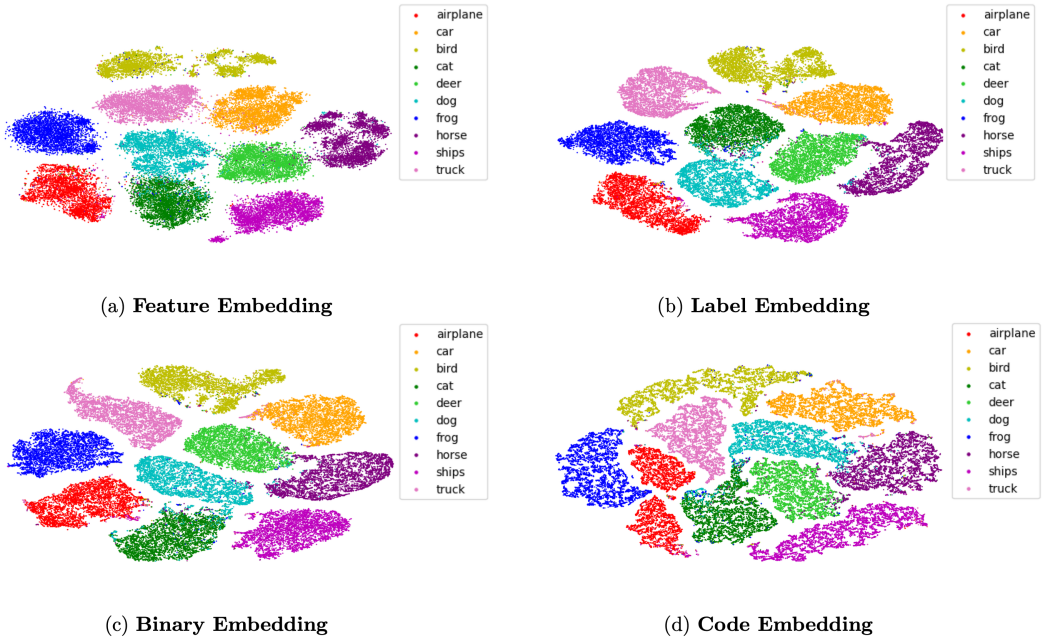
Categorizer. We use a fully connected network for this block and train it with the one-hot encoding of the labeled samples. This module receives the truncated codes to predict the labeled data. This module cannot categorize labeled data if the masker truncates too much information. Hence, it creates error signals that prevent the masker from truncating too much. This part of the network is arbitrary, and we showed in ablations that we can ignore this module without supervision signals.

2.3 Further Ablations

Feature Space Visualization. Figure 1 illustrates the tSNE visualizations for different embedding extracted from our model. While our model's features form separate clusters, our label embedding, which is the raw code feature before binarization, makes these clusters distinctive. After that, binary embedding enhances this separation while condensing the cluster by making samples of clusters closer to each other, which is evident for the bird cluster shown in yellow. Because of its 0 or 1 nature, semantic similarity will affect the binary embedding more than visual similarity. Finally, our code embedding, which assigns positional values to the extracted binary embedding, shows indirectly that to have the most efficient code, our model should span the code space as much as possible, which explains the porous nature of these clusters.

2.4 Extracting the Implicit Tree from the Model

Suppose that the generated feature vector by the network for sample X is $x_0x_1 \cdots x_k$, where k is the dimension of the code embedding or, equivalently, the depth of our implicit hierarchy tree. Using appropriate activation functions, we can assume that x_i is binary. The unsupervised contrastive loss forces the model to make the associated code to each sample unique. So if X' is not equivalent to X or one of its augmentations, its code $x'_0x'_1 \cdots x'_k$ will differ from the code assigned to X. For the supervised contrastive loss, instead of considering the code, we consider a sequence by assigning different positional values to each bit so the code $x_0x_1 \cdots x_k$ can be considered as the binary number $0.x_0x_1 \cdots x_k$. Then, the supervised contrastive loss aims to minimize the difference between these assigned binary numbers. This means our model learns to use the first digits for discriminative information while pushing the specific information about each sample to the last digits. Then, our masker learns to minimize the number of discriminative digits. Our Theorem states that, finally, the embedded tree that the model learns this way is a good approximation of the optimal tree. Ultimately, our model generates a code for each sample, and we consider each code as a binary tree traverse from the root to the leaf. Hence, the codes delineate our tree's structure and binary classification that



(a) **Feature Embedding** (b) **Label Embedding**

(c) **Binary Embedding** (d) **Code Embedding**

Figure 1: t-SNE plot for different embeddings in our model. **(a) Feature embedding.** The embedding after the projection head which is used by contrastive loss to maximize the representation information. **(b) Label embedding.** The embedding after generating code features is used by unsupervised contrastive loss for codes. **(c) Binary embedding.** The embedding by converting code features to a binary sequence using tanh activation functions and binary conditions. **(d) Code embedding.** The final truncated code which is generated by assigning positional values to the binary sequence and truncating the produced code using the masker network.

happens at each node. Since our approach enables the model to use the initial bits for supervised contrastive learning and the last bits for unsupervised contrastive learning, we can benefit from their synergic advantages while preventing them from interfering with each other.

3 Related Works

3.1 Open Set Recognition

The first sparks of the requirement for models that can handle real-world data were introduced by Scheirer et al. [20] and following works of [21, 22]. The first application of deep networks to address this problem was presented by OpenMax [23]. The main goal for open-set recognition is to distinguish *known* categories from each other while rejecting samples from *novel* categories. Hence many open-set methods rely on simulating this notion of *otherness*, either through large reconstruction errors [24, 25] distance from a set of prototypes[26–28] or by distinguishing the adversarially generated samples [29–32]. One of the shortcomings of open set recognition is that all new classes will be discarded.

3.2 Novel Class Discovery

To overcome open set recognition shortcomings, *novel class discovery* aims to benefit from the vast knowledge of the unknown realm and infer the categories. It can be traced back to [33], where they used the knowledge from labeled data to infer the unknown categories. Following this work, [34] solidified the novel class discovery as a new specific problem. The main goal of novel class discovery is to transfer the implicit category structure from the known categories to infer unknown categories [35–38, 38–57]. Despite this, the novel class discovery has a limiting assumption that test data only consists of novel categories.

3.3 Generalized Category Discovery

For a more realistic setting, *Generalized Category Discovery* considers both known and old categories at the test time. This nascent problem was introduced by [11] and concurrently under the name *open-world semi-supervised learning* by [58]. In this scenario, while the model should not lose its grasp on old categories, it must discover novel categories in test time. This adds an extra challenge because when we adapt the novel class discovery methods to this scenario, they try to be biased to either novel or old categories and miss the other group. There has been a recent surge of interest in generalized category discovery [59–73]. In this work, instead of viewing categories as an end, we investigated the fundamental question of how to conceptualize *category* itself.

3.4 Binary Tree Distillation

Benefiting from the hierarchical nature of categories has been investigated previously. Xiao [74] and Frosst and Hinton [75] used a decision tree in order to make the categorization interpretable and as a series of decisions. Adaptive neural trees proposed by [76] assimilate representation learning to its edges. Ji et al. [77] use attention binary neural tree to distinguish fine-grained categories by attending to the nuances of these categories. However, these methods need an explicit tree structure. In this work, we let the network extract this implicit tree on its own. This way, our model is also suitable when an explicit tree structure does not exist.

References

[1] Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999.

[2] Claude E Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3): 379–423, 1948.

[3] Andrei N Kolmogorov. Three approaches to the quantitative definition of information. *Problems of Information Transmission*, 1(1):1–7, 1965.

[4] Ray J Solomonoff. A formal theory of inductive inference. part i. *Information and Control*, 7(1):1–22, 1964.

[5] Ray J Solomonoff. A formal theory of inductive inference. part ii. *Information and control*, 7(2):224–254, 1964.

[6] Steven N Evans and Daniel Lanoue. Recovering a tree from the lengths of subtrees spanned by a randomly chosen sequence of leaves. *Advances in Applied Mathematics*, 96:39–75, 2018.

[7] Peter D Grünwald, Paul MB Vitányi, et al. Algorithmic information theory. *Handbook of the Philosophy of Information*, pages 281–320, 2008.

[8] Paul MB Vitányi. How incomputable is kolmogorov complexity? *Entropy*, 22(4):408, 2020.

[9] Aladin Virmaux and Kevin Scaman. Lipschitz regularity of deep neural networks: analysis and efficient estimation. *Advances in Neural Information Processing Systems*, 31, 2018.

[10] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[11] Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. Generalized category discovery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[12] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[14] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[15] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

[16] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 554–561, 2013.

[17] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3498–3505. IEEE, 2012.

[18] Kiat Chuan Tan, Yulong Liu, Barbara Ambrose, Melissa Tulig, and Serge Belongie. The herbarium challenge 2019 dataset. *arXiv preprint arXiv:1906.05372*, 2019.

[19] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

[20] Walter J Scheirer, Anderson de Rezende Rocha, Archana Sapkota, and Terrance E Boult. Toward open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1757–1772, 2012.

[21] Abhijit Bendale and Terrance Boult. Towards open world recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1893–1902, 2015.

[22] Walter J Scheirer, Lalit P Jain, and Terrance E Boult. Probability models for open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11):2317–2324, 2014.

[23] Abhijit Bendale and Terrance E Boult. Towards open set deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1563–1572, 2016.

[24] Poojan Oza and Vishal M Patel. C2ae: Class conditioned auto-encoder for open-set recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2307–2316, 2019.

[25] Ryota Yoshihashi, Wen Shao, Rei Kawakami, Shaodi You, Makoto Iida, and Takeshi Naemura. Classification-reconstruction learning for open-set recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4016–4025, 2019.

[26] Guangyao Chen, Peixi Peng, Xiangqian Wang, and Yonghong Tian. Adversarial reciprocal points learning for open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11): 8065–8081, 2021.

[27] Guangyao Chen, Limeng Qiao, Yemin Shi, Peixi Peng, Jia Li, Tiejun Huang, Shiliang Pu, and Yonghong Tian. Learning open set network with discriminative reciprocal points. In *European Conference on Computer Vision*, pages 507–522. Springer, 2020.

[28] Yu Shu, Yemin Shi, Yaowei Wang, Tiejun Huang, and Yonghong Tian. P-odn: Prototype-based open deep network for open set recognition. *Scientific reports*, 10(1):7146, 2020.

[29] ZongYuan Ge, Sergey Demyanov, Zetao Chen, and Rahil Garnavi. Generative openmax for multi-class open set classification. *arXiv preprint arXiv:1707.07418*, 2017.

[30] Shu Kong and Deva Ramanan. Opengan: Open-set recognition via open data generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 813–822, 2021.

[31] Lawrence Neal, Matthew Olson, Xiaoli Fern, Weng-Keen Wong, and Fuxin Li. Open set learning with counterfactual images. In *European Conference on Computer Vision*, pages 613–628, 2018.

[32] Zhongqi Yue, Tan Wang, Qianru Sun, Xian-Sheng Hua, and Hanwang Zhang. Counterfactual zero-shot and open-set visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15404–15414, 2021.

[33] Kai Han, Andrea Vedaldi, and Andrew Zisserman. Learning to discover novel visual categories via deep transfer clustering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8401–8409, 2019.

[34] Zhun Zhong, Enrico Fini, Subhankar Roy, Zhiming Luo, Elisa Ricci, and Nicu Sebe. Neighborhood contrastive learning for novel class discovery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10867–10875, 2021.

[35] Enrico Fini, Enver Sangineto, Stéphane Lathuilière, Zhun Zhong, Moin Nabi, and Elisa Ricci. A unified objective for novel class discovery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9284–9292, 2021.

[36] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *International Conference on Machine Learning*, pages 478–487. PMLR, 2016.

[37] Kai Han, Sylvestre-Alvise Rebuffi, Sebastien Ehrhardt, Andrea Vedaldi, and Andrew Zisserman. Automatically discovering and learning new visual categories with ranking statistics. In *Proceedings of the International Conference on Learning Representations*, 2020.

[38] Bingchen Zhao and Kai Han. Novel visual category discovery with dual ranking statistics and mutual knowledge distillation. *Advances in Neural Information Processing Systems*, 34:22982–22994, 2021.

[39] Zhun Zhong, Linchao Zhu, Zhiming Luo, Shaozi Li, Yi Yang, and Nicu Sebe. Openmix: Reviving known knowledge for discovering novel visual categories in an open world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9462–9470, 2021.

[40] KJ Joseph, Sujoy Paul, Gaurav Aggarwal, Soma Biswas, Piyush Rai, Kai Han, and Vineeth N Balasubramanian. Novel class discovery without forgetting. In *European Conference on Computer Vision*, pages 570–586. Springer, 2022.

[41] Subhankar Roy, Mingxuan Liu, Zhun Zhong, Nicu Sebe, and Elisa Ricci. Class-incremental novel class discovery. In *European Conference on Computer Vision*, pages 317–333. Springer, 2022.

[42] Mamshad Nayeem Rizve, Navid Kardan, and Mubarak Shah. Towards realistic semi-supervised learning. In *European Conference on Computer Vision*, pages 437–455. Springer, 2022.

[43] Wenbin Li, Zhichen Fan, Jing Huo, and Yang Gao. Modeling inter-class and intra-class constraints in novel class discovery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3449–3458, 2023.

[44] Muli Yang, Liancheng Wang, Cheng Deng, and Hanwang Zhang. Bootstrap your own prior: Towards distribution-agnostic novel class discovery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3459–3468, 2023.

[45] Luigi Riz, Cristiano Saltori, Elisa Ricci, and Fabio Poiesi. Novel class discovery for 3d point cloud semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9393–9402, 2023.

[46] Peiyan Gu, Chuyu Zhang, Ruijie Xu, and Xuming He. Class-relation knowledge distillation for novel class discovery. In *Proceedings of the International Conference on Learning Representations*, 2023.

[47] Yiyou Sun, Zhenmei Shi, Yingyu Liang, and Yixuan Li. When and how does known class help discover unknown ones? provable understanding through spectral analysis. In *International Conference on Machine Learning*. PMLR, 2023.

[48] Zhang Chuyu, Xu Ruijie, and He Xuming. Novel class discovery for long-tailed recognition. *arXiv preprint arXiv:2308.02989*, 2023.

[49] Colin Troisemaine, Joachim Flocon-Cholet, Stéphane Gosselin, Alexandre Reiffers-Masson, Sandrine Vaton, and Vincent Lemaire. An interactive interface for novel class discovery in tabular data. *arXiv preprint arXiv:2306.12919*, 2023.

[50] Ziyun Li, Jona Otholt, Ben Dai, Di Hu, Christoph Meinel, and Haojin Yang. Supervised knowledge may hurt novel class discovery performance. *arXiv preprint arXiv:2306.03648*, 2023.

[51] Jiaming Liu, Yangqiming Wang, Tongze Zhang, Yulu Fan, Qinli Yang, and Junming Shao. Open-world semi-supervised novel class discovery. *arXiv preprint arXiv:2305.13095*, 2023.

[52] Yuyang Zhao, Zhun Zhong, Nicu Sebe, and Gim Hee Lee. Novel class discovery in semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[53] Haoang Chi, Feng Liu, Bo Han, Wenjing Yang, Long Lan, Tongliang Liu, Gang Niu, Mingyuan Zhou, and Masashi Sugiyama. Meta discovery: Learning to discover novel classes given very limited data. In *Proceedings of the International Conference on Learning Representations*, 2022.

[54] Xinwei Zhang, Jianwen Jiang, Yutong Feng, Zhi-Fan Wu, Xibin Zhao, Hai Wan, Mingqian Tang, Rong Jin, and Yue Gao. Grow and merge: A unified framework for continuous categories discovery. In *Advances in Neural Information Processing Systems*, 2022.

[55] Qing Yu, Daiki Ikami, Go Irie, and Kiyoharu Aizawa. Self-labeling framework for novel category discovery over domains. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.

[56] KJ Joseph, Sujoy Paul, Gaurav Aggarwal, Soma Biswas, Piyush Rai, Kai Han, and Vineeth N Balasubramanian. Spacing loss for discovering novel categories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3761–3766, 2022.

[57] Ziyun Li, Jona Otholt, Ben Dai, Christoph Meinel, Haojin Yang, et al. A closer look at novel class discovery from the labeled set. *arXiv preprint arXiv:2209.09120*, 2022.

[58] Kaidi Cao, Maria Brbic, and Jure Leskovec. Open-world semi-supervised learning. In *Proceedings of the International Conference on Learning Representations*, 2022.

[59] Nan Pu, Zhun Zhong, and Nicu Sebe. Dynamic conceptional contrastive learning for generalized category discovery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[60] Sheng Zhang, Salman Khan, Zhiqiang Shen, Muzammal Naseer, Guangyi Chen, and Fahad Khan. Promptcal: Contrastive affinity learning via auxiliary prompts for generalized novel category discovery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[61] Shaozhe Hao, Kai Han, and Kwan-Yee K Wong. Cipr: An efficient framework with cross-instance positive relations for generalized category discovery. *arXiv preprint arXiv:2304.06928*, 2023.

[62] Florent Chiaroni, Jose Dolz, Ziko Imtiaz Masud, Amar Mitiche, and Ismail Ben Ayed. Parametric information maximization for generalized category discovery. *arXiv preprint arXiv:2212.00334*, 2022.

[63] Xin Wen, Bingchen Zhao, and Xiaojuan Qi. Parametric classification for generalized category discovery: A baseline study. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16590–16600, 2023.

[64] Ruoyi Du, Dongliang Chang, Kongming Liang, Timothy Hospedales, Yi-Zhe Song, and Zhanyu Ma. On-the-fly category discovery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11691–11700, 2023.

[65] Jianhong Bai, Zuozhu Liu, Hualiang Wang, Ruizhe Chen, Lianrui Mu, Xiaomeng Li, Joey Tianyi Zhou, Yang Feng, Jian Wu, and Haoji Hu. Towards distribution-agnostic generalized category discovery. In *Advances in Neural Information Processing Systems*, 2023.

[66] Sagar Vaze, Andrea Vedaldi, and Andrew Zisserman. Improving category discovery when no representation rules them all. In *Advances in Neural Information Processing Systems*, 2023.

[67] Bingchen Zhao and Oisin Mac Aodha. Incremental generalized category discovery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

[68] Bingchen Zhao, Xin Wen, and Kai Han. Learning semi-supervised gaussian mixture models for generalized category discovery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

[69] Florent Chiaroni, Jose Dolz, Ziko Imtiaz Masud, Amar Mitiche, and Ismail Ben Ayed. Parametric information maximization for generalized category discovery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1729–1739, 2023.

[70] Yanan Wu, Zhixiang Chi, Yang Wang, and Songhe Feng. Metagcd: Learning to continually learn in generalized category discovery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1655–1665, 2023.

[71] Hyungmin Kim, Sungho Suh, Daehwan Kim, Daun Jeong, Hansang Cho, and Junmo Kim. Proxy anchor-based unsupervised learning for continuous generalized category discovery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16688–16697, 2023.

[72] Wenbin An, Feng Tian, Qinghua Zheng, Wei Ding, QianYing Wang, and Ping Chen. Generalized category discovery with decoupled prototypical network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 2023.

[73] Sagar Vaze, Andrea Vedaldi, and Andrew Zisserman. No representation rules them all in category discovery. *Advances in Neural Information Processing Systems 37*, 2023.

[74] Han Xiao. Ndt: neual decision tree towards fully functioned neural graph. *arXiv preprint arXiv:1712.05934*, 2017.

[75] Nicholas Frosst and Geoffrey Hinton. Distilling a neural network into a soft decision tree. *arXiv preprint arXiv:1711.09784*, 2017.

[76] Ryutaro Tanno, Kai Arulkumaran, Daniel Alexander, Antonio Criminisi, and Aditya Nori. Adaptive neural trees. In *International Conference on Machine Learning*, pages 6166–6175. PMLR, 2019.

[77] Ruyi Ji, Longyin Wen, Libo Zhang, Dawei Du, Yanjun Wu, Chen Zhao, Xianglong Liu, and Feiyue Huang. Attention convolutional binary neural tree for fine-grained visual categorization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10468–10477, 2020.