Received 1 February 2024; revised 19 April 2024; accepted 23 April 2024. Date of publication 10 May 2024;
date of current version 1 July 2024. The review of this article was arranged by
Associate Editor Xiao Fu.

Digital Object Identifier 10.1109/OJSP.2024.3399663

Representation Synthesis by Probabilistic Many-Valued Logic Operation in Self-Supervised Learning

HIROKI NAKAMURA [1], MASASHI OKADA [1], AND TADAHIRO TANIGUCHI [1,2]

[1]Panasonic Holdings, Osaka 571-8501, Japan
[2]Ritsumeikan University, Kyoto 604-8520, Japan

CORRESPONDING AUTHOR: HIROKI NAKAMURA (email: nakamura.hiroki003@jp.panasonic.com).

ABSTRACT In this paper, we propose a new self-supervised learning (SSL) method for representations that enable logic operations. Representation learning has been applied to various tasks like image generation and retrieval. The logical controllability of representations is important for these tasks. Although some methods have been shown to enable the intuitive control of representations using natural languages as the inputs, representation control via logic operations between representations has not been demonstrated. Some SSL methods using representation synthesis (e.g., elementwise mean and maximum operations) have been proposed, but the operations performed in these methods do not incorporate logic operations. In this work, we propose a logic-operable self-supervised representation learning method by replacing the existing representation synthesis with the OR operation on the probabilistic extension of many-valued logic. The representations comprise a set of feature-possession degrees, which are truth values indicating the presence or absence of each feature in the image, and realize the logic operations (e.g., OR and AND). Our method can generate a representation that has the features of both representations or only those features common to both representations. Furthermore, the expression of the ambiguous presence of a feature is realized by indicating the feature-possession degree by the probability distribution of truth values of the many-valued logic. We showed that our method performs competitively in single and multi-label classification tasks compared with prior SSL methods using synthetic representations. Moreover, experiments on image retrieval using MNIST and PascalVOC showed the representations of our method can be operated by OR and AND operations.

INDEX TERMS Logic operation, operability, representation learning, self-supervised learning.

I. INTRODUCTION

Remarkable progress has been achieved in representation learning recently. Self-supervised learning (SSL) is a method for learning representations without labeled data [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11]. Using models pre-trained by SSL for image classification and object detection achieves high performance with a small amount of labeled data. Popular SSL methods, e.g., BYOL [2], SimSiam [1], and DINO [12], learn a model to maximize the similarity between representations of augmented views of an image. Owing to these advances in representation learning, various foundation models [13], [14], [15], [16] have been proposed. It is a pre-trained model built on large-scale datasets and can be adapted to various tasks and domains. CLIP [13] achieves zero-shot image classification by learning the common representation space of the two modalities: natural language and images. DALL-E [16] generates images by the diffusion model from a representation of text descriptions.

Control of representations, including the logic operations of representations, is important in various applications such as image generation and retrieval [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27]. Controlling representations using natural language has facilitated intuitive control of representations [13], [16], [19], [27]. However, it has been pointed out that image generation methods using natural language often inadequately reflect the logical relationships within the prompt [19], [27], [28]. Additionally, methods for controlling representations using human feedback and representation disentanglement have been proposed [20], [21], [22], [23], [24], [25], but few methods explicitly perform representation control compatible with propositional logic operations. If operations corresponding to the logical OR and AND operations could be realized, it would be possible to generate representations that combine features of both representations and representations that share only common features. We believe that achieving such logic operations between representations will expand the range of representation control and enhance the convenience of various applications utilizing representations.

In SSL, several methods use representations generated by operations between representations [29], [30], [31], [32], [33]. As representation synthesis methods, elementwise mean [30] and elementwise maximum [33] operations of two representations have been proposed. The primary objective of these methods is to learn diverse and robust features of images using mixed images [34], [35], and they have not been designed to enhance the operability of representations. As a result, these methods have not incorporate the possibility of logic operations.

In this study, we propose a logic-operable self-supervised representation learning method to enhance the operability of representations. We realize the control of representations via logic operations by applying the OR operation of the probabilistic extension of the Gödel logic [36], which is one of many-valued logic, to the representation synthesis operation of the conventional SSL methods. A representation of our method comprises degrees of feature possession, where the feature-possession degree indicates the degree to which each feature is included in the input image. Logic operations between representations can be performed by expressing the feature-possession degree in truth value. In addition, using the probabilistic extension of many-valued logic, the feature-possession degree can be presented by the categorical distribution of truth values and the expression of the ambiguous presence of features can be achieved. Fig. 1 shows examples of the feature-possession degree and the logic operation of representations. This method achieves operable representation learning by maximizing the similarity between the representation of a mixture of two images and the representation of each image synthesized by the OR operation. The contributions of this study can be summarized as follows:

- We proposed a new self-supervised representation learning method that enables logic operation between representations.
- We conducted experiments on single and multi-label classification tasks and demonstrated that our method is competitive with existing methods.
- We demonstrated that the representations successfully express the degree of feature possession without labels. In addition, the image retrieval results showed that the representation of our method is operable by OR and AND operations.

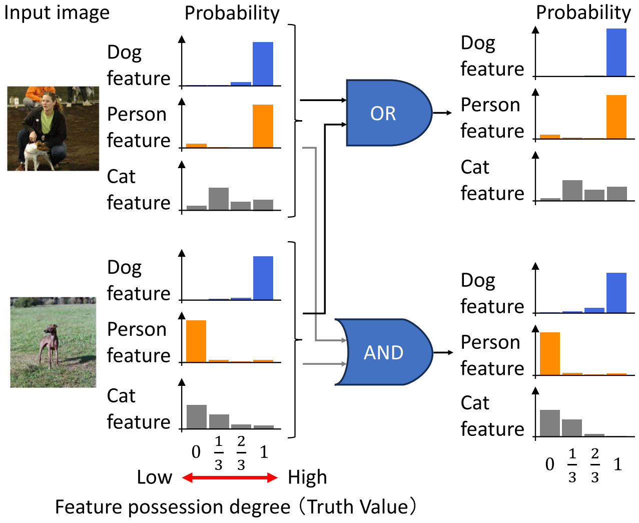


FIGURE 1. Overview of the representations and logic operation. The method predicts a representation capable of performing logic operations between representations using many-valued logic. The representation comprises several probability distributions that express the probability of the truth value of the feature-possession degree. These examples show three categorical distributions, possibly corresponding to "Dog", "Person" and "Cat." Logic operations (e.g., OR, AND) between representations can be performed by presenting feature possession in many-valued logic.

II. RELATED WORK

A. SSL AND REPRESENTATION SYNTHESIS

Previous studies have shown that SSL performs well in many downstream tasks, such as classification, object detection, and semantic segmentation [1], [2], [3], [5], [12]. SimCLR [5] and MoCo [3], called contrastive learning, learn to maximize the similarity of representation pairs augmented from the same image (*positive pairs*) and to minimize the similarity of representation pairs augmented from different images (*negative pairs*). Meanwhile, SimSiam [1], BYOL [2], and DINO [12], called non-contrastive learning, learn using only the positive pairs.

Recently, several methods [29], [30], [31], [32] that use representations synthesized from multiple representations have been proposed. Ren et al. [30] proposed SSL based on a transformer using mixed images and synthesized representations. The method improves performance by adding learning to maximize the similarity between the representations of mixed images and synthesized representations. MixSiam [33] is an SSL method that uses mixed images and representations synthesized by a maximizing operation to learn more discriminative representations. Although some SSL methods using synthesized representation have been proposed, the logic operability of representations by synthesis has not been studied in detail.

B. REPRESENTATION OPERABILITY

Representation operations and control have been studied regarding image generation, image retrieval, interpretability of representations, and generation of new representations, among others [16], [22], [23], [25], [26], [29], [30], [31], [32], [33], [37], [38]. For example, β-VAE [25] and Guided-VAE [24] improve the interpretability of representations by disentangling them, which has been found to be beneficial for both image retrieval and generation. info-GAN [26] and DALL-E [16] improve the performance of control of generated images by generating representations from controllable latent variables and languages. In addition, there are also methods [22], [23] that involve modifying representations through interactions with humans to retrieve intended images. Furthermore, i-mix [31] and MixSiam [33] realize SSL using mixed images by generating new representations upon synthesizing the representations of images. These methods enable representation learning for various features. However, they fell short of enabling logic operations between representations.

III. PRELIMINARY

A. SELF-SUPERVISED LEARNING

Our method is based on non-contrastive SSL, such as SimSiam [1] and BYOL [2], DINO [12]. The method learns representation prediction by minimizing the distance between the representations of images x_i and $x_i{}'$ obtained from the random augmentation of image X_i. Here, i indicates the image index. This method need two encoders f_θ and g_ϕ to predict image representations z_i, z_i' and $\boldsymbol{p}_i$, $\boldsymbol{p}_i'$. θ and ϕ are the parameters.

The loss $\mathcal{L}_{normal}$ is also defined by $\mathcal{D}(\cdot)$, which describe the difference between the representations, as follows;

$$\mathcal{L}_{normal} = \frac{1}{2}\mathcal{D}(\boldsymbol{p}_i, \mathrm{stopgrad}(z_i')) + \frac{1}{2}\mathcal{D}(\boldsymbol{p}_i', \mathrm{stopgrad}(z_i)). \tag{1}$$

The stop-gradient operation stopgrad$(\cdot)$ [1] prevents gradient calculation for the respective variable during backpropagation to avoid learning collapse. The model is trained by $\min_\phi \mathcal{L}_{normal}$. $\mathcal{D}(\cdot)$ differs depending on the style of representation, for example, negative cosine similarity [1], [2] and cross entropy [12].

B. SELF-SUPERVISED LEARNING WITH REPRESENTATION SYNTHESIS

This section describes a SSL method using the mixed image x_{mix} and the synthesized representation z_c. Fig. 2 shows an overview of the SSL method using mixed images and representation synthesis. Mixup [34] is used to generate the mixed image x_{mix}, z_c is generated using representations z_1 and z_2 and the synthesis function Mix$(\cdot)$. $\lambda_{mix} \in [0, 1]$ is the Mixup parameter.

$$x_{mix} = \lambda_{mix} x_1 + (1 - \lambda_{mix}) x_2, \tag{2}$$

$$z_c = \mathrm{Mix}(z_1, z_2) \tag{3}$$

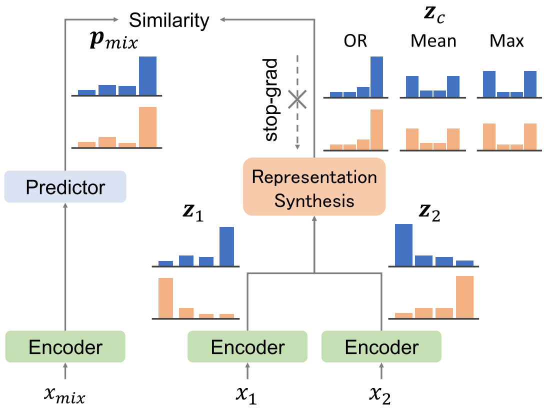


FIGURE 2. Overview of SSL using mixed images and representation synthesis, such as the logic (OR), mean, and maximum operations.

We define another loss function $\mathcal{L}_{mix}$ to minimize the distance between $\boldsymbol{p}_{mix}$, which is the representation of x_{mix} and the synthesized representation z_c. In addition, we also define the weighted average of $\mathcal{L}_{normal}$ and $\mathcal{L}_{mix}$ as the loss function $\mathcal{L}$. The model is trained by $\min_\phi \mathcal{L}$. $\alpha \in [0, 1]$ is a hyperparameter.

$$\mathcal{L}_{mix} = \mathcal{D}(\boldsymbol{p}_{mix}, \mathrm{stopgrad}(z_c)), \tag{4}$$

$$\mathcal{L} = \alpha \mathcal{L}_{normal} + (1 - \alpha)\mathcal{L}_{mix} \tag{5}$$

IV. METHOD

We propose a SSL method for logic-operable representations. First, we describe a probabilistic extension of the many-valued logic and operations in Section IV-A. Then, we describe SSL with representations which comprise multiple categorical distributions in Section IV-B. Afterward, we describe how to synthesize the representation using several operations in Section IV-C.

A. LOGIC OPERATION OF THE PROBABILISTIC MANY-VALUED LOGIC

A many-valued logic refers to a logic system with truth values other than 0 and 1. In this study, we considered Gödel logics [36]. In Gödel logics, a family G_n of many-valued logics possess n truth values $0, \frac{1}{n-1}, \frac{2}{n-1}, \ldots, 1$. Such truth values other than 0 and 1 realize the expression of uncertainty. The logic operations of variables A and B are defined as follows: A and B are the truth values, and $A, B \in [0, 1]$.

$$A \vee B := \max(A, B) \tag{6}$$

$$A \wedge B := \min(A, B) \tag{7}$$

We further define the NOT operation as $\neg A = 1 - A$.

When A and B follow categorical distributions, they are written as follows:

$$A \sim \mathrm{Cat}(A|\pi_{\mathbf{a}} = \{a_0, a_1, \ldots, a_{n-1}\}) \tag{8}$$

$$B \sim \mathrm{Cat}(B|\pi_{\mathbf{b}} = \{b_0, b_1, \ldots, b_{n-1}\}) \tag{9}$$

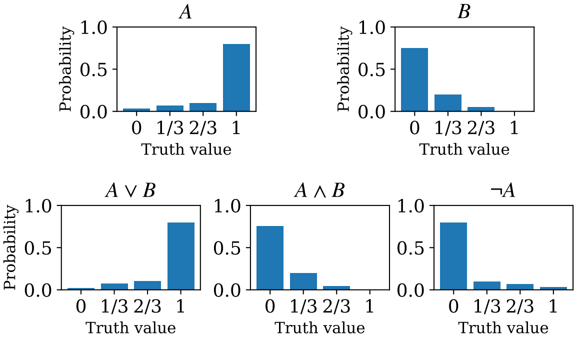


FIGURE 3. Example of the probability of each truth value for $n = 4$.

where a_i and b_i represent probabilities. We define $P(A = \frac{i}{n-1}) = a_i$ and $P(B = \frac{i}{n-1}) = b_i$.

Then, we discuss $P(A \vee B = \frac{i}{n-1})$, $P(A \wedge B = \frac{i}{n-1})$ and $P(\neg A = \frac{i}{n-1})$. They are described as;

$$P\left(A \vee B = \frac{i}{n-1}\right) = a_i \sum_{j=0}^{i} b_j + b_i \sum_{j=0}^{i} a_j - a_i b_i, \quad (10)$$

$$P\left(A \wedge B = \frac{i}{n-1}\right) = a_i \sum_{j=i}^{n-1} b_j + b_i \sum_{j=i}^{n-1} a_j - a_i b_i, \quad (11)$$

$$P\left(\neg A = \frac{i}{n-1}\right) = a_{n-1-i} \quad (12)$$

Fig. 3 shows an example of the probability of each truth value of A, B, $A \vee B$, $A \wedge B$ and $\neg A$ for $n = 4$.

B. SELF-SUPERVISED LEARNING WITH MULTIPLE CATEGORICAL DISTRIBUTIONS

In this section, we describe SSL with representations which comprise multiple categorical distributions. We prepare encoder f_θ and predictor h_ψ. The second encoder g_ϕ is defined as $g_\phi = h_\psi \circ f_\theta$. θ, ψ and ϕ are the parameters. These encoders predict representations as follows;

$$\tilde{z}_i = f_\theta(x_i),\ \tilde{z}'_i = f_\theta(x'_i), \quad (13)$$

$$\tilde{\boldsymbol{p}}_i = g_\phi(x_i),\ \tilde{\boldsymbol{p}}'_i = g_\phi(x'_i). \quad (14)$$

Representations $\tilde{z}_i$ and $\tilde{\boldsymbol{p}}_i$ comprise N M-dimensional vectors with a total number of dimensions of $N \times M$. Each vector is defined as $\tilde{z}_{i,j}$, $\tilde{\boldsymbol{p}}_{i,j}$ $(j = 1, 2, \ldots, N)$. Consequently, the vectors $\tilde{z}_{i,j}$, $\tilde{\boldsymbol{p}}_{i,j}$ are transformed into categorical distributions using the softmax function.

$$z_{i,j} = \mathrm{softmax}\big((\tilde{z}_{i,j} - c)/\tau_t\big) \quad (15)$$

$$\boldsymbol{p}_{i,j} = \mathrm{softmax}\big(\tilde{\boldsymbol{p}}_{i,j}/\tau_s\big) \quad (16)$$

$z'_{i,j}$ and $\boldsymbol{p}'_{i,j}$ are also obtained by replacing $\tilde{z}_{i,j}$ and $\tilde{\boldsymbol{p}}_{i,j}$ in (15) and (16) with $\tilde{z}'_{i,j}$ and $\tilde{\boldsymbol{p}}'_{i,j}$ respectively. Representations z_i and $\boldsymbol{p}_i$ comprise N M-dimensional categorical distributions. Thus, $z_{i,j}$, $\boldsymbol{p}_{i,j} \in \mathbb{R}^M$ and z_i, $\boldsymbol{p}_i \in \mathbb{R}^{N \times M}$. The k-th class parameter of the j-th categorical distribution is described by $z_{i,j,k}$, $p_{i,j,k} \in [0, 1]$. τ_t and τ_s are the parameters of sharpening, and c is a parameter of centering [12].

By considering cross-entropy, $\mathcal{D}(\cdot)$ in (1) is defined as follows;[1]

$$\mathcal{D}(\boldsymbol{p}_i, \mathrm{stopgrad}(z'_i)) = -\frac{1}{N} \sum_{j=1}^{N} \sum_{k=1}^{M} z'_{i,j,k} \log p_{i,j,k}. \quad (17)$$

C. REPRESENTATION SYNTHESIS

In this section, we explain the operations used to synthesize representations, Mix$(\cdot)$. First, we describe the synthesis of representations using the mean and maximum operations used in existing methods. Then, we describe representation synthesis using a many-valued logic operation to learn representations that are operable by logic operations.

1) REPRESENTATION SYNTHESIS BY THE MEAN OPERATION

This synthetic method is referenced from SDMP [30]. The synthesized representation z_c is derived from z_1, z_2, and the Mixup parameter λ_{mix}.

$$z_c = \lambda_{mix} z_1 + (1 - \lambda_{mix}) z_2 \quad (18)$$

2) REPRESENTATION SYNTHESIS BY THE MAXIMUM OPERATION

This synthetic method was referenced from MixSiam [39]. The synthesized representation z_c is the normalized elementwise maximum of the prenormalized representations $f(x_1)$ and $f(x_2)$. z_c is derived as follows, where the maximum means the elementwise maximum function;

$$z_{c,j} = \mathrm{softmax}\left(\frac{\mathrm{maximum}\big(\tilde{z}_{1,j}, \tilde{z}_{2,j}\big) - c}{\tau_t}\right). \quad (19)$$

3) REPRESENTATION SYNTHESIS BY LOGIC OPERATION

We propose a representation synthesis method based on many-valued logics. In our method, the degree of possession of each feature in the image is indicated by truth values to achieve logic operations between representations. The representation z_c in (4) is synthesized by a logic operation.

Feature-possession degree: First, we describe how to express each feature-possession degree of an image. $Z_{i,j}$ and $P_{i,j}$ are stochastic variables that follow the j-th categorical distribution of z_i, $\boldsymbol{p}_i$. When $Z_{i,j}$ and $P_{i,j}$ follow Gödel logics and have M truth values $(0, \frac{1}{M-1}, \ldots, 1)$, such as (8) and (9), $Z_{i,j}$ and $P_{i,j}$ are described as follows:

$$Z_{i,j} \sim \mathrm{Cat}(Z_{i,j} | \boldsymbol{\pi} = \{z_{i,j,1}, z_{i,j,2}, \ldots, z_{i,j,M}\}), \quad (20)$$

$$P_{i,j} \sim \mathrm{Cat}(P_{i,j} | \boldsymbol{\pi} = \{p_{i,j,1}, p_{i,j,2}, \ldots, p_{i,j,M}\}), \quad (21)$$

$$P\left(Z_{i,j} = \frac{k-1}{M-1}\right) = z_{i,j,k}. \quad (22)$$

$$P\left(P_{i,j} = \frac{k-1}{M-1}\right) = p_{i,j,k}. \quad (23)$$

[1] When $N = 1$, (17) is equivalent to that of DINO [12].

We assume that each categorical distribution $Z_{i,j}$ corresponds to a feature of image. In other words, the larger the expected value of $Z_{i,j}$, the more features of the image corresponding to $Z_{i,j}$ that it possesses. $P_{i,j}$ also has same properties.

Representation Synthesis by Logic Operation: Logic operation is possible because each feature-possession degree is expressed as a truth value. Here, we explain the OR, AND, and NOT operations. $Z_{1,j}$ and $Z_{2,j}$ are the truth values that follow the j-th categorical distribution of z_1, z_2. We set the truth values after the operation of these truth values as follows:

$$Z_{\mathrm{OR},j} := Z_{1,j} \vee Z_{2,j} \tag{24}$$

$$Z_{\mathrm{AND},j} := Z_{1,j} \wedge Z_{2,j} \tag{25}$$

$$Z_{\mathrm{NOT},j} := \neg Z_{1,j} \tag{26}$$

$Z_{1,j}$ and $Z_{2,j}$ follow categorical distributions as shown in (22). From (10), (11) and (12), $z_{\mathrm{OR},j,k}$, $z_{\mathrm{AND},j,k}$, and $z_{\mathrm{NOT},j,k}$ are as follows:

$$\begin{aligned} z_{\mathrm{OR},j,K} &= P\left(Z_{1,j} \vee Z_{2,j} = \frac{K-1}{M-1}\right) \\ &= z_{1,j,K} \sum_{k=1}^{K} z_{2,j,k} + z_{2,j,K} \sum_{k=1}^{K} z_{1,j,k} - z_{1,j,K} z_{2,j,K} \end{aligned} \tag{27}$$

$$\begin{aligned} z_{\mathrm{AND},j,K} &= P\left(Z_{1,j} \wedge Z_{2,j} = \frac{K-1}{M-1}\right) \\ &= z_{1,j,K} \sum_{k=K}^{M} z_{2,j,k} + z_{2,j,K} \sum_{k=K}^{M} z_{1,j,k} - z_{1,j,K} z_{2,j,K} \end{aligned} \tag{28}$$

$$\begin{aligned} z_{\mathrm{NOT},j,K} &= P\left(\neg Z_{1,j} = \frac{K-1}{M-1}\right) \\ &= z_{1,j,M-K+1} \end{aligned} \tag{29}$$

We define $z_1 \vee z_2$ as operations that perform (27) and $z_1 \wedge z_2$ as operations that perform (28) for all j, k in z_1, z_2. In addition, we define $\neg z_1$ as operations that perform (29) for all j, k in z_1. These operations can also be performed on $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$.

Synthesization of z_c: As the representation $\boldsymbol{p}_{mix}$ is considered to possess the features of images x_1 and x_2, the synthesized representation z_c is designed such that it has features of z_1 and z_2. To satisfy this requirement, we set $z_c = z_1 \vee z_2$.

D. EXPECTED VALUE LOSS

This section describes a new loss considering the expected value of the feature-possession degree. For example, when there are three stochastic variables of the truth value,

$$Z_{1,j} \sim \mathrm{Cat}(Z_{1,j}|\pi{=}\{z_{1,j,1} = 0.8, z_{1,j,2} = 0.1, z_{1,j,3} = 0.1\}),$$

$$P_{2,j} \sim \mathrm{Cat}(P_{2,j}|\pi{=}\{p_{2,j,1}{=}0.1, p_{2,j,2} = 0.8, p_{2,j,3} = 0.1\}),$$

$$P_{3,j} \sim \mathrm{Cat}(P_{3,j}|\pi{=}\{p_{3,j,1}{=}0.1, p_{3,j,2} = 0.1, p_{3,j,3} = 0.8\}),$$

the cross-entropy loss of $Z_{1,j}$ and $P_{2,j}$ is the same as that of $Z_{1,j}$ and $P_{3,j}$, though the expected truth values of $P_{2,j}$ and $P_{3,j}$ are different. To facilitate learning the appropriate feature-possession degrees, we introduce the new loss which penalizes more when the difference between the expected values of feature-possession degrees is large. By considering the binary cross-entropy of the expected values of the random variables $Z_{i,j}$ and $P_{i,j}$, it is described as follows;

$$\begin{aligned} &\mathrm{BCE}(\boldsymbol{p}_{i,j}, z_{i,j}) \\ &\quad = -E[Z_{i,j}] \log E[P_{i,j}] - (1 - E[Z_{i,j}]) \log (1 - E[P_{i,j}]). \end{aligned} \tag{30}$$

$E[\cdot]$ indicates the expected value. The loss functions $\mathcal{L}_{exp}$ is defined as follows;

$$\begin{aligned} \mathcal{L}_1 = \frac{1}{2N} \sum_{j=1}^{N} &(\mathrm{BCE}(\boldsymbol{p}_{i,j}, \mathrm{stopgrad}(z'_{i,j})) \\ &+ \mathrm{BCE}(\boldsymbol{p}'_{i,j}, \mathrm{stopgrad}(z_{i,j}))), \end{aligned} \tag{31}$$

$$\mathcal{L}_2 = \frac{1}{N} \sum_{j=1}^{N} \mathrm{BCE}(\boldsymbol{p}_{mix,j}, \mathrm{stopgrad}(z_{c,j})), \tag{32}$$

$$\mathcal{L}_{exp} = \alpha \mathcal{L}_1 + (1 - \alpha) \mathcal{L}_2. \tag{33}$$

Finally, we use the weighted average of $\mathcal{L}$ from (5) and $\mathcal{L}_{exp}$ as the loss function. $\beta \in [0, 1]$ is a hyperparameter.

$$\mathcal{L}_{prop} = \beta \mathcal{L}_{exp} + (1 - \beta) \mathcal{L} \tag{34}$$

$$\min_{\phi} \mathcal{L}_{prop} \tag{35}$$

V. EXPERIMENT

In this section, we experiment with the proposed method from several perspectives. First, we compare the performance of the proposed and existing methods on classification tasks. (Sections V-B, and V-C). Further, we confirm the relationship between the feature-possession degree and images and show that the feature-possession degree is learned as intended (Section V-D). In addition, we confirm the operability of the representation via logic operations using image retrieval (Section V-E).

A. IMPLEMENTATION

We implemented four methods: vanilla and representation synthesis by using the mean, maximum, and logic operations, as described in Section IV-C. Vanilla is a method that does not use mixed images or representation synthesis. Unless stated otherwise, the setup for each method is as follows.

Experimental settings for networks: The projector of the encoder f and predictor h of the proposed method are the same as those of SimSiam. The backbone network of the encoder f is Resnet-18 [40].

Experimental settings for pretraining: Unless stated otherwise, the number of categorical distributions within the representation N is 512, and the number of classes in each categorical distribution M is 8. The total number of dimensions

TABLE 1. Top-1 Accuracy of Linear-Evaluation for Each Method, Eight Experiments Were Performed, and the Mean and Standard Deviation (SD) of the Top-1 Accuracy Were Calculated

Method	ImageNet100	Cifar10
Vanilla	$72.90_{\pm 0.33}$	$89.83_{\pm 0.30}$
Mean operation [30]	$74.44_{\pm 1.64}$	$91.24_{\pm 0.32}$
Maximum operation [39]	**75.41**$_{\pm 0.85}$	$90.72_{\pm 0.18}$
Logic operation ($N = 2048,\ M = 2$)	$72.86_{\pm 2.63}$	$90.96_{\pm 0.14}$
Logic operation ($N = 1024,\ M = 4$)	$74.00_{\pm 2.74}$	$90.90_{\pm 0.72}$
Logic operation ($N = 512,\ M = 8$)	**75.41**$_{\pm 0.95}$	**91.28**$_{\pm 0.22}$
Logic operation ($N = 256,\ M = 16$)	$73.59_{\pm 0.53}$	$90.79_{\pm 0.24}$
Logic operation ($N = 128,\ M = 32$)	$73.74_{\pm 0.38}$	$90.69_{\pm 0.49}$

The results are expressed as the mean ± SD.

TABLE 2. Results of Multilabel Classification for Each Method, Eight Experiments Were Performed, and the Mean and SD of mAP Were Calculated

Method	PascalVOC	COCO
Vanilla	$61.66_{\pm 0.31}$	$43.76_{\pm 0.29}$
Mean operation [30]	$61.93_{\pm 0.56}$	$44.05_{\pm 0.33}$
Maximum operation [39]	$61.19_{\pm 0.32}$	$43.86_{\pm 0.23}$
Logic operation ($N = 2048,\ M = 2$)	$60.60_{\pm 0.99}$	$43.59_{\pm 0.89}$
Logic operation ($N = 1024,\ M = 4$)	$61.24_{\pm 1.53}$	$43.80_{\pm 1.06}$
Logic operation ($N = 512,\ M = 8$)	**62.26**$_{\pm 0.32}$	**44.59**$_{\pm 0.31}$
Logic operation ($N = 256,\ M = 16$)	$61.47_{\pm 0.50}$	$44.11_{\pm 0.33}$
Logic operation ($N = 128, M = 32$)	$61.85_{\pm 0.29}$	$43.86_{\pm 0.26}$

The results are expressed as the mean ± SD.

of the representation, $N \times M$, is 4096. We used momentum-SGD for pretraining. The learning rate was 0.027, and SGD momentum was 0.9. The learning rate followed a cosine-decay schedule. The weight decay was 1e-4. The batch size was 32. The parameters of sharpening were $\tau_s = 0.1$ and $\tau_t = 0.04$. The parameter of centering c was updated as in DINO, with the update parameter set to $m = 0.9$. The augmentation settings were the same as those used in SimSiam. The image-mixing method was Mixup [34]. The parameter for Mixup, λ_{mix}, was set to 0.5. The loss weight α in (5) and (33) is 0.5. β, the parameter for the expected value loss of the logic operation method in (34) is 0.6.

B. SINGLE-LABEL IMAGE CLASSIFICATION

We experimented with single-label image classification to verify the effectiveness of our method. First, we performed self-supervised pretraining with ImageNet100 [41][2] and CIFAR10 [42] dataset without labels to learn image representations. Then, we trained a linear classifier on frozen representations on the training set of the dataset with the labels. Finally, we calculated the top-1 accuracy of the test set and used it as an evaluation metric.

Now, we describe the experimental setup for learning a linear classifier. We used LARS [43] as an optimizer for linear evaluation. The learning rate was 1.6, and the SGD momentum was 0.9. The weight decay was 0.0, and the batch size was 512. The image augmentation was performed as in SimSiam. The number of epochs was 200. We used the model with the highest top-1 accuracy as the validation set for testing. When we trained it, we cropped an image randomly and resized it to 224 × 224 before inputting it into the model. The area of a randomly cropped image ranged from 0.08 to 1.0 of the area of the original image. In the validation and test phases, the image was resized to 256 × 256, and the center was cropped to 224 × 224.

Table 1 summarizes the results. We compared four methods: Vanilla and representation synthesis using the mean, maximum, and logic operations. Our method performs competitively compared with existing synthetic methods. In the logic operations, we compared M, the number of truth values of the many-valued logic, from 2 to 16. The results of ImageNet100 showed that the many-valued logic ($M > 2$), which can express the uncertainty of the feature-possession degree, improved accuracy compared to the binary case ($M = 2$). The performances are lower for $M = 16$ and 32 than for $M = 8$. We consider this is because if M is increased under the condition that NM is constant, N, the number of the feature-possession degree, decreases and a sufficient variety of feature-possession degrees cannot be described by a representation.

C. MULTILABEL IMAGE CLASSIFICATION

We also experimented with multilabel image classification to verify the effectiveness of our method. The model pre-trained at 200 epochs by ImageNet100 was trained on the multilabel classification dataset and evaluated in terms of mean average precision (mAP).

In the experiments, the backbone network was fixed, and only the final fully-connected layer was trained. The number of epochs was 500, and the learning rate was 0.02. When we trained the models, the input image was resized to 640 × 640, and we randomly selected {640, 576, 512, 384, 320} as the width and height to crop the images. Finally, the cropped images were resized to 224 × 224 pixels. When we evaluated the models, we resized the input image to 256 × 256 and performed center-crop operation with a size of 224 × 224.

We used Microsoft COCO [44] and PascalVOC [45] as multilabel classification datasets, which contain 80 and 20 categories, respectively. When using COCO, we trained models using the training set and evaluated them using the validation set because ground truth annotations for the test set were unavailable. When using PascalVOC, we trained models using the trainval2007 and trainval2012 sets and evaluated them using the test2007 set.

Table 2 shows the results. Our methods are competitive with other representation synthesis methods.

D. EVALUATION OF THE FEATURE-POSSESSION DEGREE

In this section, we verify that the feature-possession degree $E[P_{i,j}]$ corresponds to a certain image feature and is learned as hypothesized. Fig. 4 shows the examples of the top five images with some feature-possession degrees. We used ImageNet100 dataset. This results suggest that the feature-possession degree is learned as an indicator of how much of a certain feature an

[2] ImageNet100 is a 100-category subset of ImageNet [41].

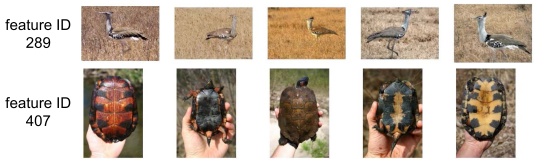


FIGURE 4. The-5 images with the highest feature-possession degrees. Feature ID is the index number of feature-possession degrees.

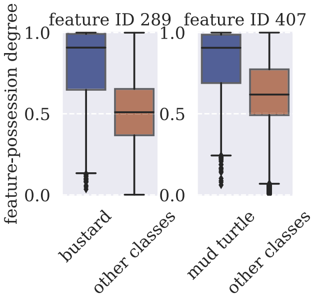


FIGURE 5. Boxplot of the feature-possession degrees of particular classes compared with others.

input image has because images with large feature-possession degrees tend to be similar.

Next, we investigated the relationship between the feature-possession degree and the image class. Fig. 5 shows the boxplot of a particular feature-possession degree in one class and the other. For example, the representation of class "bustard" images has a 289th feature-possession degree larger than the representations of the other classes. The result of the larger feature-possession degree for the images of a particular class indicates that the feature-possession degree corresponds to the image feature that is frequently included in that class.

Using a multilabel classification dataset, we also investigated the relationship between the feature-possession degree and the objects. Fig. 6 shows examples of the relationship. The left image contains two objects; the middle and right images contain one object cropped from the left image. We also show two feature-possession degrees in these images. Both feature-possession degrees are relatively high in the left image containing both objects. However, in the middle and right images, one is relatively high, and the other is relatively low, indicating there is a relationship between the objects in the image and the feature-possession degree.

E. IMAGE RETRIEVAL FROM SYNTHESIZED REPRESENTATIONS

Here, we discuss the operability between representations by performing image retrieval using MNIST and PascalVOC datasets.

1) MNIST

First, we discuss the results of image retrieval using the MNIST dataset. Fig. 7 shows the top five images of the image retrieval at each query representation. The second row

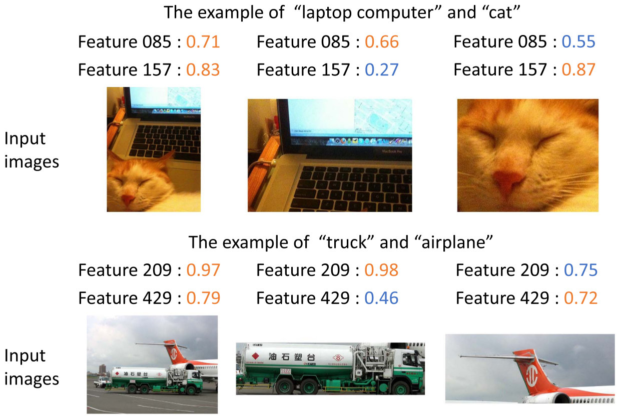


FIGURE 6. Input images and two feature-possession degrees. The left image contains two objects, both with feature-possession degrees, which are relatively high. Meanwhile, the middle and right cropped images, which have only one object, show that one feature-possession degree is relatively small.

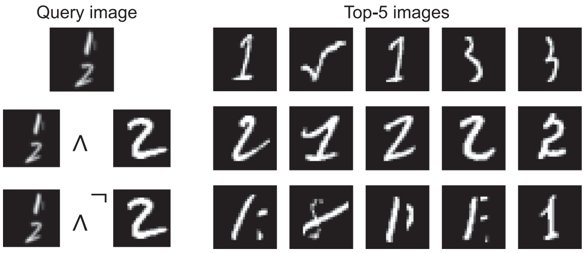
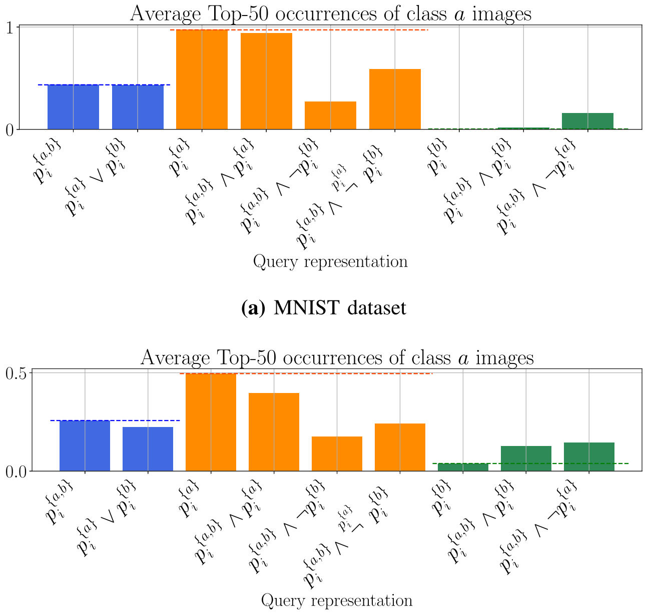


FIGURE 7. Examples of image retrieval result by the representation operated by logic operation. The top row shows images retrieved using a query for images containing both "1" and "2". The middle row shows the results of an AND operation between the representations of images containing both '1' and '2' and those containing only '2', revealing an increase in images featuring only '2' compared to the top row.

shows the result when the AND operation is performed on the representation of the image containing only "2" for the representation of the image containing both "1" and "2". Since most of the top five images are class "2" images, we assume that the "2" feature remained in the representation of the query owing to the AND operation. Conversely, when the negation of the representation of the class "2" image is performed AND operation, the features of "2" are removed from the query, indicating that the images of "1" appear more frequently in the top-5. We used the MNIST training set as the gallery set, and query images were selected from the MNIST test set. We used (17) as the distance between two representations.

In addition, we also quantitatively evaluated representation operability using an image retrieval task. Let the set of classes for MNIST be defined as L, and given classes $a, b \in L$, the sets of images in the test set that contain only class a or class b are defined as $X_{\{a\}}$ and $X_{\{b\}}$. Also, let $X_{\{a,b\}}$ denote the set of images mixup of images $x_i^{\{a\}} \in X_{\{a\}}$ and $x_j^{\{b\}} \in X_{\{b\}}$. We define $\boldsymbol{p}_i^{\{a\}}$ as the representation obtained by inputting $x_i^{\{a\}} \in X_{\{a\}}$ into the pretrained model with reference to (13),



(a) MNIST dataset

(b) PascalVOC dataset

FIGURE 8. The occurrence rate of images of class a in the top-50 when image retrieval is performed using $p_i^{\{a,b\}}$, $p_i^{\{a\}}$ and $p_i^{\{b\}}$ and the representation after logic operation of them as queries.

(14), and (16). We used the top-50 occurrence rates of the class a images as the evaluation value. First, we prepared the evaluation values when $\boldsymbol{p}_i^{\{a,b\}}$, $\boldsymbol{p}_i^{\{a\}}$, and $\boldsymbol{p}_i^{\{b\}}$ are queries, and then used these evaluation values as the reference values. Then, we prepared a representation expected to be close to the above three representations by the operations among the representations. We derived the evaluation values when the representations generated by the operations were used as queries. Operability is evaluated by comparing them with the reference values. We used the model pretrained by MNIST by 500 epochs to predict representations.

The backbone network is composed of 3 convolution layers and 2 fully-connected layers. The number of dimensions of the representation was 1024, the number of categorical distributions within the representations N was 128, and the number of classes in each categorical distribution M was 8. We randomly selected 30 $x_i^{\{a\}}$, $x_i^{\{b\}}$, and $x_i^{\{a,b\}}$ from $X_i^{\{a\}}$, $X_i^{\{b\}}$, and $X_i^{\{a,b\}}$ and show the average evaluation value when those representations are the queries. Moreover, we show the average of evaluation values of 900 combinations of two representations in operations such as $\boldsymbol{p}_i^{\{a\}} \vee \boldsymbol{p}_i^{\{b\}}$.

Fig. 8(a) shows the average of the top-50 occurrence rates. First, we compare $\boldsymbol{p}_i^{\{a,b\}}$ and $\boldsymbol{p}_i^{\{a\}} \vee \boldsymbol{p}_i^{\{b\}}$ (shown in blue). Both are expected to have features of both classes a and b. The results show that the occurrence rates are very close, indicating that the OR operation works as intended. Next, we compared $\boldsymbol{p}_i^{\{a,b\}} \wedge \boldsymbol{p}_i^{\{a\}}$ and $\boldsymbol{p}_i^{\{a,b\}} \wedge \neg \boldsymbol{p}_i^{\{b\}}$ with $\boldsymbol{p}_i^{\{a\}}$ (shown in orange), and $\boldsymbol{p}_i^{\{a,b\}} \wedge \boldsymbol{p}_i^{\{b\}}$ and $\boldsymbol{p}_i^{\{a,b\}} \wedge \neg \boldsymbol{p}_i^{\{a\}}$ with $\boldsymbol{p}_i^{\{b\}}$ (shown in green). $\boldsymbol{p}_i^{\{a,b\}} \wedge \boldsymbol{p}_i^{\{a\}}$ and $\boldsymbol{p}_i^{\{a,b\}} \wedge \neg \boldsymbol{p}_i^{\{b\}}$ would be expected to have only class a features, while $\boldsymbol{p}_i^{\{a,b\}} \wedge \boldsymbol{p}_i^{\{b\}}$ and $\boldsymbol{p}_i^{\{a,b\}} \wedge \neg \boldsymbol{p}_i^{\{a\}}$ would be expected to have class b features. The results show that $\boldsymbol{p}_i^{\{a,b\}} \wedge \boldsymbol{p}_i^{\{a\}}$ and $\boldsymbol{p}_i^{\{a,b\}} \wedge \boldsymbol{p}_i^{\{b\}}$ are very close to the reference values, indicating that the AND operation works as intended. On the other hand, $\boldsymbol{p}_i^{\{a,b\}} \wedge \neg \boldsymbol{p}_i^{\{b\}}$ and $\boldsymbol{p}_i^{\{a,b\}} \wedge \neg \boldsymbol{p}_i^{\{a\}}$ is not close to the reference values. We note that the relationship between a class and a feature-possession degree is not a direct one-to-one correspondence, but rather the class could be expressed by the presence or absence of multiple features. Even within the same class, the features may differ in terms of letter thickness, size, slant, etc. Therefore, the feature of class a could not be completely removed from $\boldsymbol{p}_i^{\{a,b\}}$, causing the occurrence rate of $\boldsymbol{p}_i^{\{a,b\}} \wedge \neg \boldsymbol{p}_i^{\{a\}}$ to be larger. In addition, it is possible that there are common features in classes a and b, and that $\boldsymbol{p}_i^{\{a,b\}} \wedge \neg \boldsymbol{p}_i^{\{b\}}$ also remove common features between the classes: thus the occurrence rates became low. Therefore, we also investigate the top occurrence rate using query $\boldsymbol{p}_i^{\{a,b\}} \wedge \overset{\boldsymbol{p}_i^{\{a\}}}{\neg}\, \boldsymbol{p}_i^{\{b\}}$, in which the features that only $\boldsymbol{p}_i^{\{b\}}$ has in $\boldsymbol{p}_i^{\{a\}}$, $\boldsymbol{p}_i^{\{b\}}$ are removed from $\boldsymbol{p}_i^{\{a,b\}}$. The operator $\overset{\boldsymbol{p}_i^{\{a\}}}{\neg}\, \boldsymbol{p}_i^{\{b\}}$ is defined as follows:

$$\overset{\boldsymbol{p}_i^{\{a\}}}{\neg}\, \boldsymbol{p}_i^{\{b\}} := \neg \boldsymbol{p}_i^{\{b\}} \vee (\boldsymbol{p}_i^{\{a\}} \wedge \boldsymbol{p}_i^{\{b\}}) \tag{36}$$

Since the common feature $\boldsymbol{p}_i^{\{a\}} \wedge \boldsymbol{p}_i^{\{b\}}$ is not removed from $\boldsymbol{p}_i^{\{a,b\}}$, the value of $\boldsymbol{p}_i^{\{a,b\}} \wedge \overset{\boldsymbol{p}_i^{\{a\}}}{\neg}\, \boldsymbol{p}_i^{\{b\}}$ is closer to the reference value than $\boldsymbol{p}_i^{\{a,b\}} \wedge \neg \boldsymbol{p}_i^{\{b\}}$.

2) PASCALVOC

Futher, we evaluated the operability of the representation on PascalVOC, a multilabel dataset of natural images. We defined L as the set of classes with at least 200 images containing only one class from the trainval2007 and trainval2012 sets. In addition, we defined sets of images in test2007 set that contain only class $a \in L$ or class $b \in L$ are defined as $X_{\{a\}}$ and $X_{\{b\}}$. We also define a set $X_{\{a,b\}}$ of images in test2007 set that contains only class a and class b. We prepared the model pretrained by trainval2007 and trainval2012 sets of PascalVOC by 500 epochs. A gallery set of 2400 images was created by sampling 200 images from the trainval2007 and trainval2012 sets for each class. Using the classes, image sets, and models defined above, we compared the occurrence rates of class a in similar experiments in Section V-E-1

Fig. 8(b) shows the average occurrence rate. Similar to the MNIST results, we can confirm that OR and AND are working properly by comparing $\boldsymbol{p}_i^{\{a,b\}}$ to $\boldsymbol{p}_i^{\{a\}} \vee \boldsymbol{p}_i^{\{b\}}$, $\boldsymbol{p}_i^{\{a\}}$ to $\boldsymbol{p}_i^{\{a,b\}} \wedge \boldsymbol{p}_i^{\{a\}}$, and $\boldsymbol{p}_i^{\{b\}}$ to $\boldsymbol{p}_i^{\{a,b\}} \wedge \boldsymbol{p}_i^{\{b\}}$. The difference from the reference value appears to be larger than that of MNIST. The reason may be that they are likely to have different features within the same class (e.g., gender and age for the "person" class, and breed for the "dog" class).

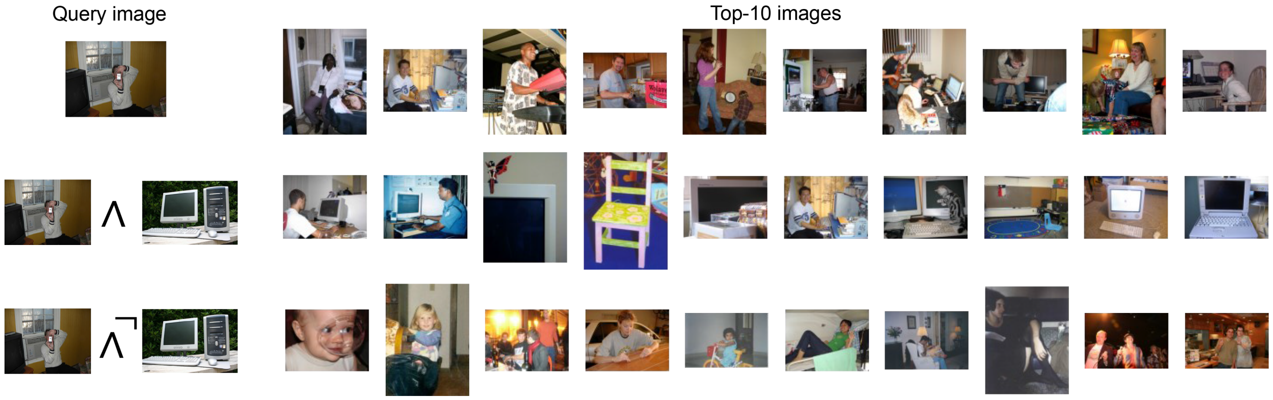


FIGURE 9. Examples of the top-10 images of image retrieval. The top row displays the images retrieved using a query for an image containing both 'tvmonitor' and 'person'. The middle row presents results from an AND operation between images containing both 'tvmonitor' and 'person' and those containing only 'tvmonitor'. This shows an increase in images containing only 'tvmonitor' compared to the top row. The results in the bottom row are derived from an AND operation between the representation of images containing both 'tvmonitor' and 'person' and the negation of the representation of images containing only 'tvmonitor'. This operation has resulted in the removal of the 'tvmonitor' feature, and an increase in images that contain only 'person'.

Fig. 9 shows some examples of image retrieval, including the result of AND and NOT operations between the representation of the image of "tvmonitor" and the representation of the image containing both "tvmonitor" and "person". In the result of second row, many images of "tvmonitor" appear in the top-10 images, indicating that the representation possesses features related to "tvmonitor" owing to the AND operation. On the other hand, in the third row, the AND, NOT operation removes the features related to "tvmonitor", and many "person" images appear in the top-10 images.

VI. LIMITATION

The computational cost of our proposed representation synthesis by logic operations is higher than that by mean and maximum operations. When a representation consists of N M-class categorical distributions, the computational cost required for synthesis is $O(NM)$ for the mean and maximum operations and $O(NM^2)$ for the logic operations. We note that our method increases the computational cost, especially in the synthesis of the representations, but this increase does not affect the gradient computation during the learning of the encoder part. Although there is an increase in computational cost, this increase is minimal.

This study focused on only image representation. However, if the operability discussed in this study can be extended to multimodal representation learning using images and language in the future, the controllability of representation can be improved.

VII. CONCLUSION

In this study, we proposed a self-supervised representation learning method that can perform logic operations by describing the degree of possession of each feature in an image using many-valued logic. The results showed that the performance of our method was competitive with those of other SSLs that use representation synthesis in classification tasks. Further, we experimentally verified that the possession degree of each feature in the image was properly learned. Moreover, we showed that the proposed method can learn representations with high logic-operability using image retrieval tasks.

REFERENCES

[1] X. Chen and K. He, "Exploring simple siamese representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 15750–15758.

[2] J.-B. Grill et al., "Bootstrap your own latent-a new approach to self-supervised learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 21271–21284.

[3] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9729–9738.

[4] X. Chen, S. Xie, and K. He, "An empirical study of training self-supervised vision transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 9640–9649.

[5] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1597–1607.

[6] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny, "Barlow twins: Self-supervised learning via redundancy reduction," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 12310–12320.

[7] Y. Tian, X. Chen, and S. Ganguli, "Understanding self-supervised learning dynamics without contrastive pairs," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 10268–10278.

[8] A. Newell and J. Deng, "How useful is self-supervised pretraining for visual tasks?," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 7345–7354.

[9] N. Komodakis and S. Gidaris, "Unsupervised representation learning by predicting image rotations," in *Proc. Int. Conf. Learn. Representations*, 2018.

[10] M. Noroozi and P. Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 69–84.

[11] H. Nakamura, M. Okada, and T. Taniguchi, "Representation uncertainty in self-supervised learning as variational inference," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 16484–16493.

[12] M. Caron et al., "Emerging properties in self-supervised vision transformers," in *Proc. IEEE Int/CVF. Conf. Comput. Vis.*, 2021, pp. 9650–9660.

[13] A. Radford et al., "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 8748–8763.

[14] M. Oquab et al., “Dinov2: Learning robust visual features without supervision,” 2023, *arXiv:2304.07193*.

[15] T. Brown et al., “Language models are few-shot learners,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 1877–1901.

[16] A. Ramesh et al., “Zero-shot text-to-image generation,” in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 8821–8831.

[17] S. Dey, P. Riba, A. Dutta, J. Llados, and Y.-Z. Song, “Doodle to search: Practical zero-shot sketch-based image retrieval,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2179–2188.

[18] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, “Hierarchical text-conditional image generation with clip latents,” 2022, *arXiv:2204.0612*.

[19] K. Sueyoshi and T. Matsubara, “Predicated diffusion: Predicate logic-based attention guidance for text-to-image diffusion models,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024.

[20] M. Havaei, X. Mao, Y. Wang, and Q. Lao, “Conditional generation of medical images via disentangled adversarial inference,” *Med. Image Anal.*, vol. 72, 2021, Art. no. 102106.

[21] A. Gonzalez-Garcia, J. V. D. Weijer, and Y. Bengio, “Image-to-image translation for cross-domain disentanglement,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2018.

[22] X. Guo, H. Wu, Y. Cheng, S. Rennie, G. Tesauro, and R. Feris, “Dialog-based interactive image retrieval,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2018.

[23] M. Levy, R. Ben-Ari, N. Darshan, and D. Lischinski, “Chatting makes perfect: Chat-based image retrieval,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2024, pp. 61437–61449.

[24] Z. Ding et al., “Guided variational autoencoder for disentanglement learning,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 7920–7929.

[25] I. Higgins et al., “beta-VAE: Learning basic visual concepts with a constrained variational framework,” in *Proc. Int. Conf. Learn. Representations*, 2016.

[26] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, “Infogan: Interpretable representation learning by information maximizing generative adversarial nets,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2016.

[27] R. Rassin, E. Hirsch, D. Glickman, S. Ravfogel, Y. Goldberg, and G. Chechik, “Linguistic binding in diffusion models: Enhancing attribute correspondence through attention map alignment,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2023, pp. 3536–3559.

[28] T. H. Truong, T. Baldwin, K. Verspoor, and T. Cohn, “Language models are not naysayers: An analysis of language models on negation benchmarks,” in *Proc. 12th Joint Conf. Lexical Comput. Semantics*, 2023, pp. 101–114.

[29] Z. Shen, Z. Liu, Z. Liu, M. Savvides, T. Darrell, and E. Xing, “Un-mix: Rethinking image mixtures for unsupervised visual representation learning,” in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 2216–2224.

[30] S. Ren et al., “A simple data mixing prior for improving self-supervised learning,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 14595–14604.

[31] K. Lee, Y. Zhu, K. Sohn, C.-L. Li, J. Shin, and H. Lee, “i-Mix: A domain-agnostic strategy for contrastive representation learning,” in *Proc. Int. Conf. Learn. Representations*, 2021.

[32] S. Kim, G. Lee, S. Bae, and S.-Y. Yun, “Mixco: Mix-up contrastive learning for visual representation,” 2020, *arXiv:2010.06300*.

[33] Y. Zhang, Y. Yin, Y. Zhang, and R. Zimmermann, “Mix-up self-supervised learning for contrast-agnostic applications,” in *Proc. IEEE Int. Conf. Multimedia Expo*, 2022, pp. 1–6.

[34] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, “mixup: Beyond empirical risk minimization,” in *Proc. Int. Conf. Learn. Representations*, 2018.

[35] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, “Cutmix: Regularization strategy to train strong classifiers with localizable features,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 6023–6032.

[36] K. Godel, “Zum intuitionistischen aussagenkalkul,” *Anzeiger Akademie der Wissenschaften Wien, Mathematisch-Naturwissenschaftliche Klasse*, vol. 69, pp. 65–66, 1932.

[37] Y. Sun, J. Zeng, S. Shan, and X. Chen, “Cross-encoder for unsupervised gaze representation learning,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 3702–3711.

[38] M. Mirza and S. Osindero, “Conditional generative adversarial nets,” 2014, *arXiv:1411.1784*.

[39] X. Guo, T. Zhao, Y. Lin, and B. Du, “Mixsiam: A mixture-based approach to self-supervised representation learning,” 2021, *arXiv:2111.02679*.

[40] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[41] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.

[42] A. Krizhevsky et al., “Learning multiple layers of features from tiny images,” Technical report, University of Toronto, 2009.

[43] Y. You, I. Gitman, and B. Ginsburg, “Large batch training of convolutional networks,” 2017, *arXiv:1708.03888*.

[44] T.-Y. Lin et al., “Microsoft coco: Common objects in context,” in *Proc. 13th Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.

[45] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (VOC) challenge,” *Int. J. Comput. Vis.*, vol. 88, pp. 303–338, 2010.