Classification-Driven Discrete Neural Representation Learning for Semantic Communications

Wenhui Hua, *Student Member, IEEE*, Longhui Xiong, Sicong Liu, *Senior Member, IEEE*, Lingyu Chen, *Member, IEEE*, Xuemin Hong, *Member, IEEE*, João F. C. Mota, *Member, IEEE*, and Xiang Cheng, *Fellow, IEEE*

***Abstract*—Semantic communications is a key enabler of the Internet of Things (IoT). By focusing on the semantic meaning of data rather than bit-level recovery, it allows intelligent agents to communicate necessary information at much lower rates. A promising technique for semantic communications is *discrete neural representation learning* (DNRL). The main idea is to learn discrete symbols from low-level, high dimensional sensory data, such that each symbol is grounded to a meaningful pattern in the sensory domain. This paper proposes a DNRL scheme that integrates three mechanisms into a coherent framework: contrastive learning, sparse coding, and neural index quantization. The proposed scheme is applied to public image datasets for lossy image compression with a downstream classification task. Results show that the proposed approach produces a highly compact continuous latent representation and a semantic discrete representation, with marginal degradation to the classification accuracy. The interpretability and consistency of the learned sub-symbolic discrete representations are validated by experiments of neural-net dissection, neural-net visualization, and MaxAmp-K classification test, a concept that we propose to evaluate classification performance of extremely compressed signals. Finally, the discrete representations are shown to be useful in rate-adaptive distributed sensing applications at the low-to-medium signal-to-noise ratios (SNR).**

***Index Terms*—Data compression, distributed detection, image classification, image representations, neural networks, quantization**

I. Introduction

SEMANTIC communications represents a paradigm shift in communications. Rather than aiming at perfect transmission of bits as in conventional communications, semantic communications focus on reliable transmission of semantic or task-relevant content. A key driving force underlying such a paradigm shift is the proliferation of machine intelligence in numerous applications such as consumer robotics, intelligent transportation, and smart manufacturing [1]–[3]. Most of these applications are rooted in the Internet of Things (IoTs), and are characterized by vast amounts of sensory data, which often has to be parsimoniously represented for cost reduction, energy efficiency, or better interpretability. These emerging demands have motivated a recent interest in the field of semantic communications [4]–[11].

Although there are several different definitions of semantic communications [3]–[14], they all agree on its goal: both the sender and receiver exploit prior knowledge about the context of the communication, for example, its semantic meaning or knowledge about the end-task. This enables the encoder to embed meaningful content into the messages, and the decoder to distill such meaning from them. Compared to the traditional Shannon-paradigm of communications, semantic communications exhibits two distinct features that make it particularly valuable for IoT. First, semantic encoding generates messages with significantly lower rates compared to the original data, allowing for more efficient utilization of network resources. The messages can also be prioritized based on their contextual importance, which improves power efficiency, reliability, and latency. When combined with adaptive channel coding and resource scheduling techniques, it also increases coverage [15]. Second, the encoded messages embedded with semantic meaning can be associated to a knowledge system, which enables automated inference and easy human interpretation. Such an integration of semantic communication and a knowledge system promises intelligent, trusted network management and service orchestration for IoT systems [16].

A key component in any semantic communication system is a discrete symbolic system, where each symbol is embedded with semantic meanings. There are two main approaches for confering semantic meaning to symbols. One approach defines symbols as abstract patterns/features in high-dimensional, continuous-valued data, which is prevalent in the physical sensing of natural signals. Finding a mapping between symbols and sensory data is commonly referred to as the symbol grounding problem [17], [18]. The second approach defines symbols by relating them to other symbols, just like a dictionary defines words using other words. Both classic [19]–[24] and recent studies [4], [7]–[9] in semantic communications have predominantly focused on the second approach, i.e., ungrounded semantics. However, the symbol grounding problem remains relatively under-explored in the context of semantic communications. This problem is important for two reasons. First, a major viewpoint in epistemology asserts that knowl-

Wenhui Hua is with the School of Electronic Science and Engineering and the National-Local Joint Engineering Research Center of Navigation and Location Services, Xiamen University. Longhui Xiong, Sicong Liu, Lingyu Chen, and Xuemin Hong are with the School of Informatics and the National-Local Joint Engineering Research Center of Navigation and Location Services, Xiamen University, Xiamen 361000, China. Sicong Liu is also with the National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China (e-mail: huawenhui@stu.xmu.edu.cn; xionglonghui@stu.xmu.edu.cn; liusc@xmu.edu.cn; chenly@xmu.edu.cn; xuemin.hong@xmu.edu.cn)(Corresponding author: Xuemin Hong).

João F. C. Mota is with the School of Engineering and Physical Sciences, Heriot-Watt University, Edinburgh EH14 4AS, U.K. (e-mail: j.mota@hw.ac.uk).

Xiang Cheng is with the State Key Laboratory of Advanced Optical Communication Systems and Networks, School of Electronics, Peking University, Beijing 100871, P. R. China (e-mail: xiangcheng@pku.edu.cn).

edge is primarily rooted in sensory experience. This viewpoint gives fundamental importance to symbol grounding as the main route for true semantics. Second, symbol grounding is closely intertwined with lossy compression, a central topic in digital communication systems.

In recent years, neural representation learning (NRL) has emerged as a powerful technique to tackle the symbol grounding problem. A wealth of NRL techniques have been proposed in the literature. Depending on whether or not semantically labeled datasets are used during neural-net training, these techniques can be categorized into supervised [25] or unsupervised [26] NRLs. Supervised techniques are in general more tractable than unsupervised ones due to the additional information provided by the labels. Our paper addresses the problem of semantic representation learning by means of supervised discrete neural representation learning (DNRL).

Problem statement: We aim to explore whether and how deep neural networks (DNNs) can be designed to solve the symbol grounding problem. Specifically, this is known as the problem of discrete neural representation learning (DNRL). By using image datasets as a representative and intuitive source of sensory data, we train a DNN to learn discrete representations (i.e., semantic symbols) of images. The semantic symbols should enable highly accurate image classification while simultaneously being highly compact (and thus efficiently compressable). Additionally, the discrete symbols should exhibit certain semantic properties, including semantic alignment [27]–[30], logical compositionality [31], [32], and interpretability.

To contrast our work with recent standardization efforts, for example, AI-based novel image coding (JPEG-AI) [33], and video for machine (VCM) [34], we note that the former mainly aims at high-fidelity reconstruction of images, while the latter considers both reconstruction and task-relevant feature extraction. Our work does not aim at perfect reconstruction of the source, but seeks to obtain highly compact and semantically interpretable representations that are useful for distributed sensing and classification tasks. Therefore, the proposed method, when applied to image data, can be seen as a candidate branch of the VCM standard and a complementary extension to the JPEG-AI standard.

Our main contributions are summarized as follows.

- We propose a new neural network design that yields highly compact and effective continuous representations. This is achieved via a novel mask-based sparsity mechanism, which we show is essential for transforming traditionally highly distributed neural representations into semantically compact representations.
- We propose novel neural quantization schemes that effectively preserve semantic information during the quantization process while achieving ultra-low and progressively adaptable coding rates.
- We study the effectiveness with which semantic information is embedded into a representation. To do so, we develop a novel test called MaxAmp-K classification test, which evaluates how semantic information is preserved under extreme compression. This test, along with other studies of neural network dissection and visualization, shows that the proposed sparse contrastive neural network learns compact and semantically interpretable representations.
- We demonstrate that the learned semantic representations, which are highly compact and yet robust to noise and interference, are particularly useful in distributed sensing applications.

The remainder of the paper is organized as follows. Section II introduces a continuous neural representation learning scheme, which lays a foundation for the discrete neural representation learning scheme proposed in Section III. Experimental results and discussions are provided in Section IV. Finally, conclusions are drawn in Section V.

II. Continuous representation learning

The discrete neural representation learning (DNRL) problem consists of designing DNNs that output discrete semantic meaningful representations. However, using discrete variables in DNNs has proven challenging due to their discontinuity and thus undefined gradients [35]. To tackle this problem, DNRL typically consists of two phases. The first phase learns a good continuous representation as the baseline, followed by a second phase of discretization and refinement.

This section focuses on the first phase, continuous neural representation learning in a supervised setting. In particular, we consider a labeled image dataset and a classification task. The problem of image semantic representation has been investigated for high-level tasks such as event understanding [36], [37], in which classification is a fundamental function. We introduce two independent mechanisms that contribute to learning better semantic representations: supervised contrastive learning and mask-based sparse coding. These two mechanisms are then compared with their classic counterparts, namely cross-entropy based supervised learning and feature-based sparse coding, respectively.

A. Supervised training and contrastive learning

We consider two paradigms for the supervised training of a DNN. The first is classic end-to-end supervised learning, which uses cross-entropy as a loss function. The second is supervised contrastive learning.

1) Classic end-to-end supervised learning: This approach focuses on the output of the task network and optimizes all of its sub-components based on the final output. As a result, intermediate representation layers and final classifier layers are optimized jointly.

Let $\mathbf{M} = \{m_i, \overline{m}_i\}_{i=1}^{N}$ represent a batch of data consisting of N images $m_i \in \{0, \ldots, 255\}^{3\times H\times W}$ and respective labels $\overline{m}_i \in [0,1]^O$, where $H \times W$ are the dimensions of the images (with 3 color channels), and O the number of classes. The semantic encoder (representation learning module) is denoted as $\boldsymbol{S}_{\boldsymbol{\alpha}}(\cdot)$, while the semantic decoder (classification module) is denoted as $\boldsymbol{S}_{\boldsymbol{\gamma}}^{-\mathbf{1}}(\cdot)$. The encoder encodes each input sample m into a feature vector $\mathbf{X} = \boldsymbol{S}_{\boldsymbol{\alpha}}(m)$, which is subsequently used by the semantic decoder to generate the categorical output $\overline{m'} = \boldsymbol{S}_{\boldsymbol{\gamma}}^{-\mathbf{1}}(\mathbf{X})$.

The cross entropy loss function is defined as

$$\mathcal{L}_{ce} = -\tfrac{1}{N}\sum_{i=1}^{N} \overline{m_i}\log(\overline{m_i'}) + (1-\overline{m_i})\log(1-\overline{m_i'}). \quad (1)$$

2) Supervised contrastive learning: Supervised contrastive learning is a recent paradigm for representation learning [29]. The main idea is to align the representation vectors of semantically-similar data in latent space. For an input batch of data $\mathbf{M}$, data augmentation techniques are first applied to obtain two similar copies of the batch. The data augmentation function is denoted as $\mathrm{Aug}(\cdot)$. Both copies are then propagated forward through the encoder network (denoted as $\boldsymbol{S_\alpha}(\cdot)$) to obtain a normalized embedding. During training, this representation is further propagated through a projection network [28], [29], which projects the data onto a lower dimensional space to speed up training. The projection head is denoted as $\mathrm{proj}(\cdot)$. The supervised contrast loss is calculated based on the output of the projection network. However, during inference, once the network is trained, the projection network is discarded.

For each input sample m, random samples $m^{'}$=Aug(m) are generated through the data augmentation function. The encoder encodes the sample $m^{'}$ into a feature vector $\mathbf{X} = \boldsymbol{S_\alpha}(m^{'})$, and the projection head encodes the feature into a vector $\boldsymbol{z}$=proj($\mathbf{X}$). Given a batch of N raw images and corresponding labels $\{m_i, \overline{m_i}\}_{i=1}^{N}$, the output of data augmentation consists of $2N$ pairs of images, which are used for training.

When the network implemented by the encoder is a CNN, the feature map it computes is a tensor $\mathbf{X} \in \mathbb{R}^{C\times H\times W}$ with C channels and spatial dimensions H and W (height and width). Thus, the total feature size is $N_f = C \times H \times W$. These features are the input to the task network, e.g., classification. In a communication scenario, the encoder is located at the transmitter end, while the task network is located at the receiver end.

The supervised contrastive learning (SCL) loss function is [29]

$$\mathcal{L}_{scl} = \sum_{i\in\mathbf{M}} \frac{-1}{|\boldsymbol{\mathcal{H}}(i)|} \sum_{h\in\boldsymbol{\mathcal{H}}(i)} \log \frac{\exp(\boldsymbol{z}_i \cdot \boldsymbol{z}_h/\tau)}{\sum\limits_{a\in\boldsymbol{\mathcal{A}}(i)} \exp(\boldsymbol{z}_i \cdot \boldsymbol{z}_a/\tau)}, \quad (2)$$

where $\cdot$ represents the dot product, τ is a temperature hyperparameter, and $\boldsymbol{\mathcal{A}}(i) \equiv \mathbf{M}\backslash\{i\}$. The index i is called the anchor, $\boldsymbol{\mathcal{H}}(i)\equiv \{h \in \boldsymbol{\mathcal{A}}(i) : \overline{m_h} = \overline{m_i}\}$ is the set of indices of all images in $\mathbf{M}$ with the same label as image m_i (but excluding i) and $|\boldsymbol{\mathcal{H}}(i)|$ is its cardinality.

As (2) imposes constraints on the latent space representations, the representations learned by SCL perform well across several downstream tasks. For example, in a classification task, the classifier should be trained separately from the encoder. This can be done by fixing the SCL encoder and training the classifier using cross-entropy (1).

3) Approximation and interpretation of SCL loss: A simple framework for contrastive learning was originally proposed in [28] and its underlying mechanisms were studied in [30]. SCL, as an extension of contrastive learning, was initially proposed in [29] to handle labelled data as additional information. We now offer an intuitive interpretation of the SCL loss based on an approximation of the log-sum-exp (LSE) function. The LSE function is defined as $\mathrm{LSE}(x_1, x_2, \ldots, x_N) = \log \sum_{i=1}^{N} \exp x_i$ and is a differentiable approximation to the maximum function. That is, for any $t > 0$,

$$\begin{aligned} \tfrac{1}{t}\,\mathrm{LSE}(tx) &> \max\{x_1, \ldots, x_N\} \\ \tfrac{1}{t}\,\mathrm{LSE}(tx) &\le \max\{x_1, \ldots, x_N\} + \tfrac{\log(N)}{t}. \end{aligned} \quad (3)$$

Applying the above inequalities to (2) yields

$$\Gamma < \mathcal{L}_{scl} < \Gamma + \log(2N), \quad (4)$$

where

$$\begin{aligned} \Gamma &= \frac{1}{\tau} \sum_{i\in\mathbf{M}} \frac{1}{|\boldsymbol{\mathcal{H}}(i)|} \sum_{h\in\boldsymbol{\mathcal{H}}(i)} \left[\max_{a\in\boldsymbol{\mathcal{A}}(i)} (\boldsymbol{z}_i \cdot \boldsymbol{z}_a) - \boldsymbol{z}_i \cdot \boldsymbol{z}_h \right] \\ &= \frac{1}{\tau} \sum_{i\in\mathbf{M}} \left[\max_{a\in\boldsymbol{\mathcal{A}}(i)} (\boldsymbol{z}_i \cdot \boldsymbol{z}_a) - \frac{1}{|\boldsymbol{\mathcal{H}}(i)|} \sum_{h\in\boldsymbol{\mathcal{H}}(i)} \boldsymbol{z}_i \cdot \boldsymbol{z}_h \right]. \end{aligned} \quad (5)$$

The term containing the max operator computes the minimum angle between an anchor and its nearest neighbour (Note that the latent vectors $\boldsymbol{z}$ is normalized to locate on a hypersphere), while the right term represents the average angle between an anchor and other samples within the same class. Minimizing the left term encourages all the latent vectors to be as spread out (over the unit hypersphere) as possible. Maximizing the right term, in turn, encourages the latent vectors associated to the images of the same class to be as close as possible. The combined effect of the two terms produces the overall dispersive yet semantically clustered representations in the latent space.

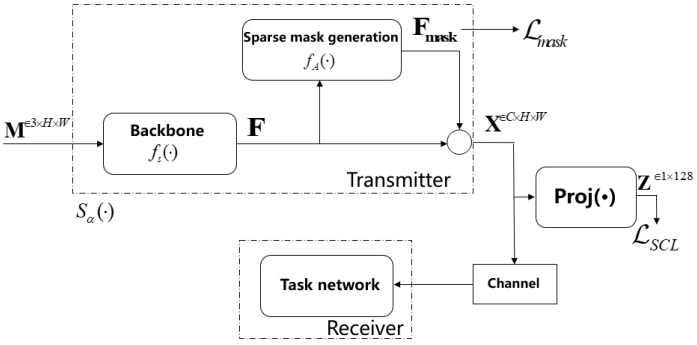


Fig. 1. Architecture of neural representation learning with mask-based sparsity constraint. ∘ stands for the Hadamard product.

B. Sparse coding

Neural networks tend to learn distributed representations, in the sense that features in deeper layers capture abstract concepts. Adding sparsity constraints has the potential to improve semantic representation learning by encouraging the network to transform low-level features into high-level abstract features. Sparse coding techniques have been effectively applied to auto-encoders and variational auto-encoders (VAE) [38] as a soft bottleneck of the neural network. In this paper, we consider three different types of sparsity constraints.

1) Feature-based sparse coding: Classic methods directly impose sparsity on the features extracted by a DNN backbone, using the ℓ_1-norm as a tractable approximation to the ℓ_0-norm. Specifically, if $\mathbf{F}$ denotes the features extracted by a DNN backbone, the sparsity loss is

$$\mathcal{L}_{\text{feature}} = ||\mathbf{F}||_1, \tag{6}$$

where the ℓ_1-norm applies to the vectorization of $\mathbf{F}$, i.e., it is the sum of the absolute values of all of its entries.

2) Mask-based sparse coding: We propose an alternative method, which we name *mask-based sparse coding*. The main idea is illustrated in Fig. 1, in which an additional branch is added to the network (for convenience, called mask-net) to compute a mask based on each input feature. Here, input data is denoted as $\mathbf{M}$. The DNN backbone, denoted as $f_s(\cdot)$, maps the input data into a feature vector $\mathbf{F} = f_s(\mathbf{M}) \in \mathbb{R}^{C\times H\times W}$. The proposed mask-net, denoted as $f_A(\cdot)$, outputs a feature-dependent mask given by $\mathbf{F}_{\text{mask}} = f_A(\mathbf{F}) \in \mathbb{R}^{C\times H\times W}$. The final output that is fed into to the classifier is $\mathbf{X}= \mathbf{F} \circ \mathbf{F}_{\text{mask}}$, where $\circ$ stands for Hadamard product. We train the DNN backbone $f_s(\cdot)$ and the mask generating network $f_a(\cdot)$ jointly.

In contrast with feature-based sparse coding, we impose sparsity on the mask, rather than on the features:

$$\mathcal{L}_{\text{mask}} = ||\mathbf{F}_{\text{mask}}||_1. \tag{7}$$

The proposed mask-based sparse coding yields a different behavior feature-based sparse coding. In particular, when $f_A(\cdot)$ is a simple network with limited capacity, the network tends to reduce the overall dimension of the effective features by using a small masking window. This is because a simple network struggles to capture the complex variations of the low-level features and selects instead the most informative features. Thus, it tends to learn a fixed-size masking window with small dimensions. This window gradually eliminates unimportant semantic representations over the course of training and forces the representation to focus on a small number of "effective features". According to the auto-encoder literature [39], there are two common approaches to design bottleneck layers. One is to limit the number of hidden units in an intermediate layer, while the other is to impose a sparsity constraint on a large number of units. The proposed mask-based sparsity technique is thus a combination of both approaches, blending sparse feature selection and dimensionality reduction.

It can also be argued that, compared with feature based sparse coding using ℓ_1-norm, the proposed mask-based method provides a better approximation to the ℓ_0-norm of $\mathbf{X}$ because it does not directly depend on the magnitudes of $\mathbf{X}$.

3) Weighted sparse coding: It is possible to combine the above two sparse coding techniques to yield a new loss, called weighted sparse coding, given by

$$\mathcal{L}_{\text{ws}} = ||\mathbf{F}_{\text{mask}} \circ \mathbf{F}||_1. \tag{8}$$

(8) with properly selected weights $\mathbf{F}_{\text{mask}}$. To motivate this new loss, it is important to note that, theoretically, a weighted ℓ_1-norm $\|\mathbf{F}_{\text{mask}} \circ \mathbf{F}\|_1$ with appropriately selected weights $\mathbf{F}_{\text{mask}}$ can approximate the ℓ_0-norm $\|\mathbf{F}\|_0$ better than a simple ℓ_1-norm $\|\mathbf{F}\|_1$ [40]–[42]. Indeed, in the sparse approximation literature [40]–[42], the optimal weights are a function of the optimal vector/matrix: $(\mathbf{F}^\star_{\text{mask}})_{ij} = 1/\mathbf{F}^\star_{ij}$. Our approach, which computes the weights $\mathbf{F}_{\text{mask}} = f_A(\mathbf{F})$ as a function of $\mathbf{F}$, can then be interpreted as approximating a weighted ℓ_1-norm.

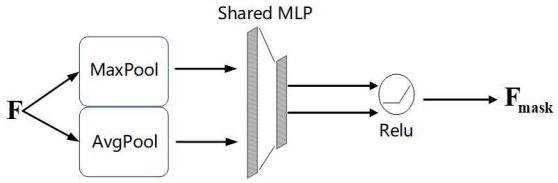


Fig. 2. Implementation of the mask-net for sparsity constraint.

4) Implementation of mask-net: Different types of neural networks can be implemented as the mask-net. Here, we adopt a network with a structure similar to the one in [43]–[45], as illustrated in Fig. 2. Our structure draws on the strengths of the previous work by using two down-sampling structures in parallel. Two tensors, $\mathbf{F}_{\text{avg}} \in \mathbb{R}^{C\times 1\times 1}$ and $\mathbf{F}_{\text{max}} \in \mathbb{R}^{C\times 1\times 1}$ are generated by performing maximum and average pooling on $\mathbf{F} \in \mathbb{R}^{C\times H\times W}$, respectively. A shared MLP layer is then used to reduce the dimension of $\mathbf{F}_{\text{avg}}$ and $\mathbf{F}_{\text{max}}$ to $\mathbb{R}^{C/2\times H\times W}$. Finally, the mask is obtained by concatenating the two MLP outputs and applying the ReLU activation function.

5) Gradient analysis: The underlying differences of the above three sparse coding techniques can be clarified by performing a gradient analysis on the sparsity loss terms with respect to the final layer of output latent representation. Note that for feature-based sparsity coding, the final output is $\mathbf{F}$, as there is no mask-net. For the other two sparsity codings, the final output is $\mathbf{X}$. We have

$$\frac{\partial \mathcal{L}_{\text{feature}}}{\partial \mathbf{F}} = \text{sign}(\mathbf{F}) \tag{9}$$

$$\begin{aligned}\frac{\partial \mathcal{L}_{\text{mask}}}{\partial \mathbf{X}} &= \frac{\partial \mathcal{L}_{\text{mask}}}{\partial \mathbf{F}} \cdot \frac{\partial \mathbf{F}}{\partial \mathbf{X}} \\ &= \frac{\text{sign}\left(f_A(\mathbf{F})\right) \circ \frac{\partial f_A(\mathbf{F})}{\partial \mathbf{F}}}{\frac{\partial f_A(\mathbf{F})}{\partial \mathbf{F}} \circ \mathbf{F} + f_A(\mathbf{F})}\end{aligned} \tag{10}$$

$$\frac{\partial \mathcal{L}_{\text{ws}}}{\partial \mathbf{X}} = \text{sign}(\mathbf{X}). \tag{11}$$

Note that the entries of $\mathbf{F}$ and $\mathbf{X}$ are non-negative due to the ReLU functions before the output of backbone and mask-net. As a result, the partial derivative of the feature sparsity and weighted sparsity tends to be uniform across all active dimensions. On the contrary, mask-based sparsity tends to produce uneven derivatives. Such a flexibility is beneficial for sparsity purpose as activation trade-offs are allowed across different dimensions.

C. Combination of methods

The above sparsity coding and supervised learning methods can be used in combination, leading to

$$\mathcal{L}oss = \mathcal{L}_{basic} + \lambda \cdot \mathcal{L}_{sparse}, \tag{12}$$

where $\mathcal{L}_{basic}$ can either be the contrastive learning loss defined in (2) or the cross-entropy loss in (1), and $\mathcal{L}_{sparse}$ can

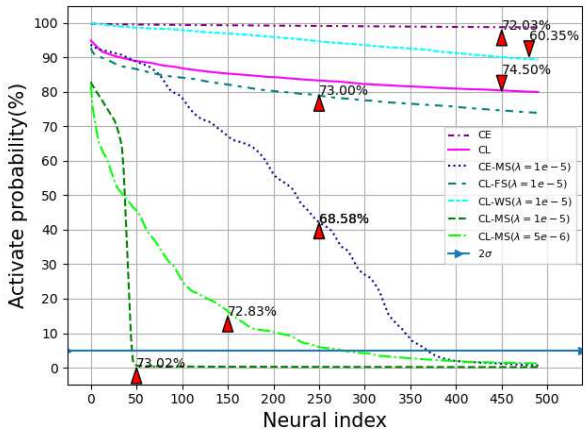


Fig. 3. Ordered distribution of neuron activation probability (CIFAR100 data set). The triangle represents the precision of each loss, while the 2σ line indicates the probability of activation exceeding the 2σ confidence level.

Algorithm 1 Training of CE-MS

Input: A batch of data $\mathbf{M}$

Output: Encoder network parameters α, classifier network parameters $\boldsymbol{\gamma}$

// Training

while Stopping criterion is not met **do**
  Sample a batch of data $\mathbf{m}$
  Encoder encodes $\mathbf{m}$ into $\mathbf{X}$. Classifier decodes $\mathbf{X}$ into Target
  Calculate loss $\mathcal{L}_{CE-MS}$ by (12)
  Update $\boldsymbol{\alpha} \longleftarrow$ Gradient descent ($\boldsymbol{\alpha}$, $\mathcal{L}_{CE-MS}$)
  Update $\boldsymbol{\gamma} \longleftarrow$ Gradient descent ($\boldsymbol{\gamma}$, $\mathcal{L}_{CE-MS}$)
end while
// Finish training
Return the parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$

either be the feature-based sparsity defined in (6), mask-based sparsity in (7) or weighted sparsity in (8). The performances of different combinations will be evaluated and compared in Section V. Here, λ is a hyper-parameter that controls the weight of sparsity constraints. The backbone DNN can have an arbitrary structure, e.g., VGG [46], Resnet [47], or Transformer [48], etc. For mask-based sparse coding, we train the backbone DNN and mask-net jointly. The training procedures of cross-entropy learning with mask-based sparsity (CE-MS) and contrastive learning with mask-based sparsity (CL-MS) are described in Algorithms 1 and 2, respectively. The stopping criterion on both algorithms is based on the loss function not decreasing enough for a few epochs. Specifically, the training stops whenever the loss does not decrease during five consecutive epochs.

To further illustrate the degree of sparsity in learned representations, Fig. 3 shows the ordered neural activation probabilities of different schemes and for different values of λ. Specifically, it shows in order of decreasing magnitude the active (i.e., non-zero output) probability of each neuron over the CIFAR100 validation set. We can see that the proposed mask-based sparsity mechanism can effectively reduce the

Algorithm 2 Two-stage Training of CL-MS

Input: A batch of data $\mathbf{M}$

Output: Encoder network parameters $\boldsymbol{\alpha}$, classifier network parameters $\boldsymbol{\gamma}$

// Encoder training

while Stopping criterion is not met **do**
  Sample a batch of data $\mathbf{m}$
  Encoder encodes $\mathbf{m}$ into $\mathbf{X}$.
  Calculate loss $\mathcal{L}_{CL-MS}$ by (12)
  Update $\boldsymbol{\alpha} \longleftarrow$ Gradient descent ($\boldsymbol{\alpha}$, $\mathcal{L}_{CL-MS}$)
end while
// Finish training
//Classifier network training
while Stopping criterion is not met **do**
  Freeze encoder parameters $\boldsymbol{\alpha}$
  Classifier decodes $\mathbf{X}$ into Target
  Calculate loss $\mathcal{L}_{CE}$ by (1)
  Update $\boldsymbol{\gamma} \longleftarrow$ Gradient descent ($\boldsymbol{\gamma}$, $\mathcal{L}_{CE}$)
end while
// Finish training
Return the parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$

number of active neurons, leading to a compact representation via "effective features". Moreover, the hyperparameter λ can control the degree of sparsity in CL-MS.

III. Discrete Neural Representation Learning

Discrete representation learning is a crucial step towards our goal of semantic symbolic grounding. There are two common paradigms for quantization: scalar quantization [49]–[51] and vector quantization [35], [52], [53]. Scalar quantization is typically ineffective in terms of compression performance [11]. Vector quantization requires building an extra dictionary, which may undermine the semantic aspect of the representation generated by the encoder. In particular, disentangled latent representations may become entangled. We propose a new neural quantization scheme called *neural index quantization*, which not only requires fewer bits than scalar quantization, but also has better semantic properties.

A. Neural index quantization

As shown in Fig. 3, the integration of SCL and mask-based sparsity leads to very compact continuous representations. This motivates us to propose a class of quantization schemes called neural index quantization, which consist of two steps. The first step is to select a subset of K useful neural indices based on a certain criteria, followed by a second step of conventional scalar quantization on the selected indices. The output of other neurons are discarded and assigned zero value by default. Two criteria for index selection are proposed below.

1) MaxPro-K quantization: This scheme selects the K neuron indices that have the highest activation probability over the training data set. The underlying assumption is that semantic information of the entire data set is mostly preserved in a small number of highly-active neurons. In other words,

we apply a fixed mask to $\mathbf{X}$, according to the activation pattern over a test dataset. More precisely, we compute the set

$$\mathcal{K} = \{i \mid P(i) > \chi, i \in \{1, 2, \ldots, N_f\}\}, \tag{13}$$

where $P(i)$ is the empirical activation probability of the ith neuron over the tested dataset, χ is a predefined threshold, and $K = |\mathcal{K}|$. Taking the results shown in Fig. 3 for example, at $\chi = 95.45$ (i.e., the 2σ percentile), the value of K for CL-MS (λ = 1e-5) and CE-MS (λ = 1e-5) is 48 and 374, respectively. Clearly, this scheme is only useful when the continuous representations are compact.

2) MaxAmp-K quantization: This scheme selects the K neuron indices that have the largest output amplitude for a given input. The underlying assumption is that semantic-information of an input is mostly preserved in a small number of active neurons with large outputs. In practice, K can be as small as one. This scheme is therefore useful when the continuous representations are sparse. We note that in our context, sparsity and compactness are two different concepts. The former refers to the total number of active neurons for a fixed input, while the latter refers to the number of highly active neurons over a dataset.

B. Existing evidence for neural index quantization

The neural index quantization schemes proposed above rely on the assumption that the semantic information of a dataset can be preserved by a number of highly active neurons. This assumption is supported by a wide range of studies including artificial neural network dissection (e.g., [54]) and bio-inspired neural networks such as spiking neural networks (SNNs) [55]. These studies have consistently demonstrated that specific neurons within a neural network often encode distinct and semantically meaningful information.

Neural network dissection studies have shown that in well-trained artificial neural networks such as CNNs, certain neurons may exhibit a high degree of specialization (e.g., [54]). These specialized neurons tend to respond selectively to specific features or concepts in the input data. In essence, they effectively encode and preserve specific semantic information.

In addition, bio-inspired SNNs (e.g., [55]), promoted as the third-generation artificial neural networks, also provide compelling empirical support for the assumption. Inspired by the functioning of the brain, SNNs build on spiking neurons and temporal coding. They perform tasks such as classification by having a small set of neurons represent complex information via temporal firing patterns.

In summary, the assumption on which neural index quantization relies, i.e., that semantic information in a dataset can be captured by a small number of highly active neurons, is well supported both by studies of conventional neural networks and by SNNs.

C. Bandwidth analysis

We can measure the average bandwidth of neural quantization schemes as bits per input sample. Bandwidth is characterized by three parameters: the total number of dimensions N_f, the number of selected indices K, and the number of quantization bits n.

Assuming a 32-bit floating point representation, the number of bits (bandwidth) for a learned continuous representation is

$$B_{\text{continuous}} = N_f \cdot 32. \tag{14}$$

For neural index quantization, the bandwidth is

$$B_{\text{neural}} = \log_2 \mathcal{C}_{N_f}^K + K \cdot n, \tag{15}$$

where the first term corresponds to $\log_2 \mathcal{C}_{N_f}^K$ is the number of bits required to encode the K (out of N_f) selected indices. The second term, which represents the number of bits required to encode each dimension, scales linearly with K.

IV. Experiments

We now describe experiments designed to evaluate the performance of the proposed representation learning schemes. We focus on classification tasks, which involve the control of semantic compression and transmission [56], [57].

To achieve this, we adopt convolutional neural networks as the underlying DNN backbone. Specifically, we implement $f_s(\cdot)$ in Fig. 1 with a ResNet50 [47], which outputs a vector representation with size 2048, i.e., $H = W = 1$ and $C = N_f = 2048$. Three public datasets (MNIST, CIFAR10, CIFAR100 [58]) are used for supervised representation learning and image classification tasks.

For the method employing the cross-entropy (CE) loss, the initial learning rate is set to 0.02, and the batch size to 64. For the contrastive learning (CL) method, these two parameters are set to 0.05 and 32, respectively, and the temperature parameter τ is set to 0.07. We use the SGD optimizer with a momentum parameter of 0.9 and weight decay $1e^{-4}$.

A. Performance of CL-MS

We perform an ablation study by using the six methods mentioned in Section II, namely cross-entropy learning (CE), contrastive learning (CL), cross-entropy learning with mask-based sparsity (CE-MS), contrastive learning with mask-based sparsity (CL-MS), contrastive learning with weighted sparsity (CL-WS) and contrastive learning with feature-based sparsity (CL-FS). Three performance metrics are considered. Apart from classification accuracy, we introduce two additional metrics to assess the sparsity of the learned representations. The first metric is N_1, the number of features that become active at the 2σ confidence level over all tested samples, which corresponds to the number of features with an activation probability greater than 0.045 over the entire test set. In other words, this metric measures the overall compactness of the representation. The second metric is $\overline{N_2}$, which is the average number of features activated by a single sample.

Table. I shows the results. We make the following observations. First, CL outperforms CE in classification accuracy by a small margin in all the three datasets. This validates CL as an effective representation learning method. However, vanilla CL and CE all have large numbers of active features $N_1(2\sigma)$ and $\overline{N_2}$, which means they learn highly distributed representations that defy semantic interpretation.

TABLE I
EXPERIMENTAL RESULTS OF CONTINUOUS REPRESENTATION LEARNING METHODS

DataSets	Methods	$N_1(2\sigma)$	$\overline{N_2}$	Accuracy
MNIST	CE	2048	2030.50	99.50%
MNIST	CL	2048	2028.60	99.56%
MNIST	CL-FS($\lambda = e^{-5}$)	1884	1415.50	99.21%
MNIST	CL-WS($\lambda = e^{-5}$)	136	59.65	99.35%
MNIST	CE-MS($\lambda = e^{-5}$)	26	12.69	99.48%
MNIST	CL-MS($\lambda = e^{-5}$)	2	2.00	**99.56**%
CIFAR10	CE	2048	2032.60	95.00%
CIFAR10	CL	2048	1958.40	96.00%
CIFAR10	CE-MS($\lambda = e^{-5}$)	138	65.85	92.50%
CIFAR10	CL-FS($\lambda = e^{-4}$)	1102	347.56	94.90%
CIFAR10	CL-FS($\lambda = e^{-5}$)	2048	136.50	95.10%
CIFAR10	CL-FS($\lambda = e^{-6}$)	2048	1723.56	94.88%
CIFAR10	CL-WS($\lambda = e^{-5}$)	434	121.26	90.50%
CIFAR10	CL-MS($\lambda = e^{-4}$)	4	2.29	94.81%
CIFAR10	CL-MS($\lambda = 5e^{-5}$)	8	5.36	95.00%
CIFAR10	CL-MS($\lambda = e^{-5}$)	16	11.20	**95.25**%
CIFAR10	CL-MS($\lambda = e^{-6}$)	234	116.19	95.00%
CIFAR100	CE	2048	1952.05	72.03%
CIFAR100	CL	2048	1527.96	74.50%
CIFAR100	CL-FS($\lambda = e^{-5}$)	2048	1314.41	73.00%
CIFAR100	CL-WS($\lambda = e^{-5}$)	1936	1284.26	60.35%
CIFAR100	CE-MS($\lambda = e^{-5}$)	374	312.20	68.58%
CIFAR100	CL-MS($\lambda = e^{-5}$)	48	33.60	**73.02**%

Second, CL-MS outperforms CL-FS in the CIFAR10 dataset, with varying values of sparsity parameter λ. Both methods achieve similar accuracies. However, the compactness and sparsity performance of CL-MS is significantly better than CL-FS in all datasets. This validates the advantage of the proposed mask-based sparsity method over the conventional feature-based sparsity method. As expected, sparsity performance improves with increasing λ. As the difficulty of the task increases, i.e., when the dataset becomes more complex (going from MNIST, to CIFAR10, to CIFAR100), the performance of CL-WS gradually deteriorates, which is evident in the observed decline of its performance.

Third, CL-MS is shown to outperform CE-MS in all the datasets. CL-MS not only has better accuracy, but also achieves much better performance in compactness and sparsity. This means that the proposed combination of contrastive learning and mask-based sparsity concentrates semantic information into a compact representation.

Fourth, the value of hyperparameter λ, which weighs the sparsity constraint, has a direct impact on the performance of CL-MS. The experiments on CIFAR10 show that the compactness of representations increases monotonically with increasing λ. This corroborates the neuron activation probability results shown in Fig. 3. However, in practice, the neural network may not converge if λ is too large. On the other hand, regarding classification accuracy, there is an optimal value of λ. In Table I, the best accuracy of CL-MS on CIFAR10 is given by λ=1e-5, outperforming two other benchmarks at λ=5e-5 and λ=1e-6 by 0.25%. This suggests that in practice, the value of λ could be empirically optimized given targeted bandwidth (related to representation compactness) and task performance (e.g., classification accuracy). To this end, automatic DNN hyperparameter tuning methods proposed in the literature [59], such as Bayesian model based approach [60] or grid search approach [61], can be potentially applied.

TABLE II
THE COMPUTATIONAL PARAMETERS AND FLOPS

Methods	Network parameters (M)	GFLOPs
CL	23.5	2.62
CE	23.5	2.62
CL-FS	23.5	2.62
CL-MS	25.6	2.63
CE-MS	25.6	2.63

Overall, if we compare CL-MS and CL quantitatively, we can see that the dimension of useful representations (i.e., effective features) is reduced by a factor around 100, at a cost of reducing the classification accuracy by 0.75 %. The highly compact representation of CL-MS translates into significant gains in bandwidth in a communication system.

B. Computational complexity

The computational complexity of an encoder can be measured in terms of the number of network parameters and giga floating-point operations (GFLOPs). The complexity of CE, CL, CE-MS, CL-MS, and CL-FS encoders used in our experiments are compared in Table. II. We can see that mask-based sparsity only incur marginal increase in complexity compared to other baselines. In practice, neural network models are first trained on a specific dataset in an offline fashion. The resulting network is then deployed and can be further optimized by fine-tuning network parameters online.

C. Bandwidth and classification accuracy with different representations

Table. III compares the classification accuracy of continuous and quantized representations. All discrete quantizations are based on the same continuous representation learning scheme CL-MS with H=W=1(Dimensions of the output representations). For the scalar quantization, we apply the straight-through estimator (STE) technique as introduced in [50] and use n=3 bits for each scalar value. For the MaxPro quantization, we set K=$N_1(2\sigma)$=48 based on the probabilistic interpretation as described by Eqn (13). Each of the K selected dimensions is then quantized with full accuracy of 32 bits. For the MaxAmp scheme, we set K=3 experimentally, a value that yielded satisfactory accuracy performance. We note that smaller value at K=1,2 are later tested in Table. IV to investigate the performance under extreme compression. Finally, the MaxProAmp scheme inherits the same parameter settings of MaxPro and MaxAmp. The CIFAR100 dataset is used for classification. We can see that with comparable bandwidth, the accuracy of MaxPro is slightly worse than that of the scalar quantization scheme, by 0.22%. MaxAmp, on the other hand, achieves a reduction in bandwidth by a factor over 50-fold, with only 1.77% decrease in accuracy. Finally, in the combined MaxProAmp scheme, the accuracy decreases by 7.16%, but the bandwidth is reduced by nearly 100 times.

TABLE III
EXPERIMENT RESULTS ON BANDWIDTH AND LINEAR CLASSIFIER PERFORMANCE ANALYSIS

Methods	Bandwidth estimation formula	Accuracy(%)	$N_1(2\sigma)$	n	Actual bandwidth
CL	$N_1(2\sigma)$*32	74.50	2048	32 bits	65,536 bits
Scalar quantization	$N_1(2\sigma)$*n	73.27	578	3 bits	1734 bits
Max Pro(K=48)	$\log_2 \mathcal{C}^K_{N_1(2\sigma)} + K * n$	73.05	48	32 bits	1536 bits
Max Amp(K=3)	$\log_2 \mathcal{C}^K_{N_1} + K * n$	71.50	2048	1 bit	34 bits
Max ProAmp(K=3)	$\log_2 \mathcal{C}^K_{N_1(2\sigma)} + K * n$	66.11	48	1 bit	18 bits

TABLE IV
EXPERIMENTAL RESULTS ON MAXAMP-K CLASSIFICATION TEST

Methods	Naive Bayesian(%)	Decision Tree(ID3)(Maxdepth=21,80)(%)	Linear Classifier(%)	SVM(%)	Random Forest(%)
	$K = (1, 2, 3)$	$K = (1, 2, 3)$	$K = (1, 2, 3)$	$K = (1, 2, 3)$	$K = (1, 2, 3)$
CL-MS($\lambda = e^{-5}$)	36.66 64.92 **70.00**	18.15(36.66) 57.58(**68.30**) 66.92(**69.85**)	36.50 **68.80 71.50**	36.70 **68.31 71.05**	36.47 **68.49 70.96**
CL-MS($\lambda = 5e^{-6}$)	**58.99 65.93** 66.41	16.18(**51.69**) 36.29(65.59) 45.07(65.81)	**59.02** 67.15 68.23	**64.41** 66.80 67.02	**64.44** 67.07 67.63
CE-MS($\lambda = e^{-5}$)	22.71 38.36 45.68	10.93(20.89) 19.72(36.28) 26.29(40.16)	22.80 35.62 45.67	29.94 42.76 45.19	29.93 40.74 43.00
CL	50.12 65.65 69.17	11.53(28.14) 18.51(46.06) 24.37(52.80)	48.39 61.36 69.24	39.57 64.88 68.20	50.55 64.31 67.57
CE	25.61 41.09 48.80	11.14(22.11) 21.24(37.22) 27.02(40.68)	25.78 39.27 48.02	25.66 40.37 47.72	25.63 39.57 45.52

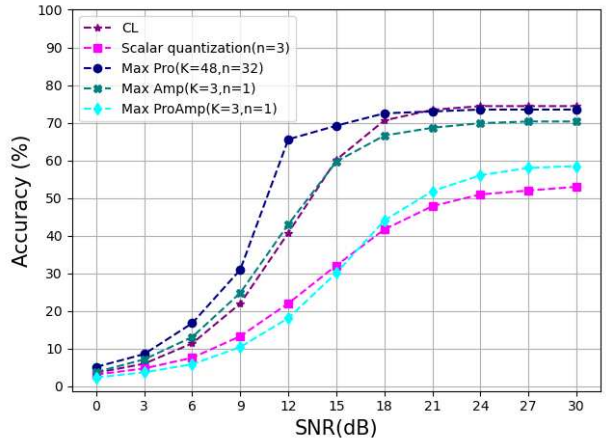


Fig. 4. Classification accuracy as a function of the SNR of latent representations.

D. Effects of noise impairment

In a communications scenario, noise and distortion during transmission may impair the encoding of semantic information. Fig. 4 investigates the reliability of semantic representation subject to noise. Our experiment consists of three steps: 1) Encoders and classifiers for different coding schemes are trained without noise on the CIFAR100 dataset; 2) Gaussian white noise is introduced into the vector representations of different coding schemes. For quantization schemes, the vector takes discrete values. For example, for 1-bit quantization ($n = 1$), each entry of the vector is either 0 or 1. The power of the vector is normalized to 1, and additive Gaussian white noise with varying signal-to-noise ratio (SNR) is applied. 3) The impaired vector representation is used to evaluate classification accuracy. This experiment thus simulates random impairments at the representation level.

Fig. 4 shows that the MaxPro quantization scheme has the best performance in terms of noise resilience, followed by MaxAmp, which shows almost the same performance as CL. On the contrary, performance of the traditional scalar quantization scheme ($n = 3$) is much worse compared with the two proposed neural index quantization schemes. As the proposed schemes yield sparser representations, these results imply that semantic information is more robust to noise than conventional, distributed representations.

E. Test on different classifiers for interpretability

So far, we have demonstrated that the proposed neural network designs effectively learn a compact and discrete representation. In practice, the representation learning network can be deployed as the source encoder at the transmitter end, while the classification network can be deployed as the decoder at the receiver end. The latent vector, or learned representation, is the information that should be transmitted across the channel. A compact representation is obviously beneficial in this scenario, as lower bit rates can be readily converted into performance gains in power consumption [62], link reliability [63], and coverage [64].

We propose a test called MaxAmp-K classification test for quantitative evaluation of semantic embedding quality in different representations. Our test includes three simple steps. The first step is to apply 1-bit quantization to the representation under test. Therefore the original representation is converted into a binary representation. The second step is to train a classifier on that binary representation to perform the original classification task. In particular, decision trees and naive Bayesian classifiers are preferred because they are highly interpretable. A simple linear classifier can also be used for performance benchmark. Finally, the trained classifier is applied to samples in the test set and the classification accuracy is measured.

The MaxAmp-K test is based on the hypothesis that the activation of a feature unit corresponds to the observation of a useful conceptual entity, or symbol. This hypothesis is supported by a number of empirical studies [55], [65]. Moreover, the combination composition of the observed symbols can be used to make one-step logical (binary) inference on the label.

Table. IV shows the results of the above tests for CE, CL, CE-MS and CL-MS using the CIFAR100 dataset.

Fig. 5. The activation pattern of different feature units to different categories.

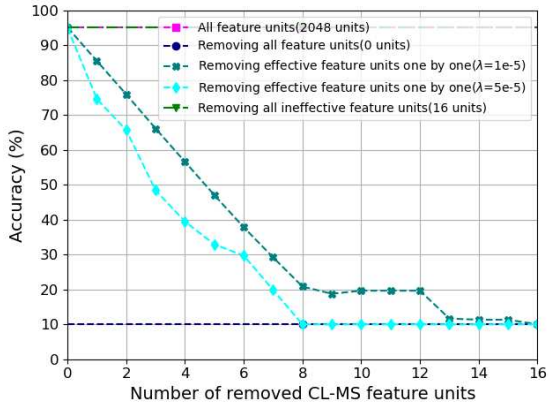


Fig. 6. Test accuracy after sequential removal of semantic symbols.

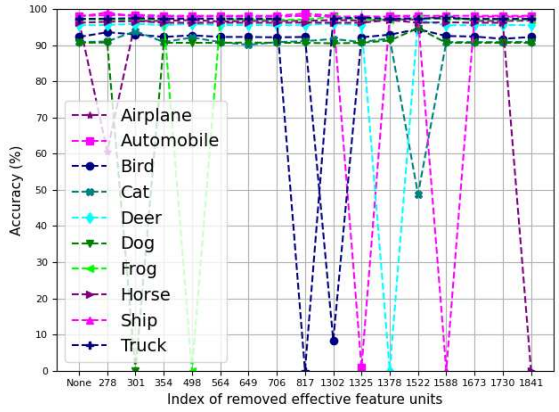
We have the following observations. First, CL significantly outperforms CE in all types of classifiers, indicating the fundamental importance of contrastive learning to learn semantically-meaningful representations. Second, CL-MS outperforms CL on linear classifiers, Naive Bayesian, decision trees (Maxdepth=21,80), support vector machines (SVM), and random forests (Subtrees=100). Moreover, when λ is properly selected, CL-MS achieves better performance than CL for small values of K. This shows that the proposed sparsity mechanism also has a positive role in encouraging semantic embedding.

F. Visualization of spatial attention

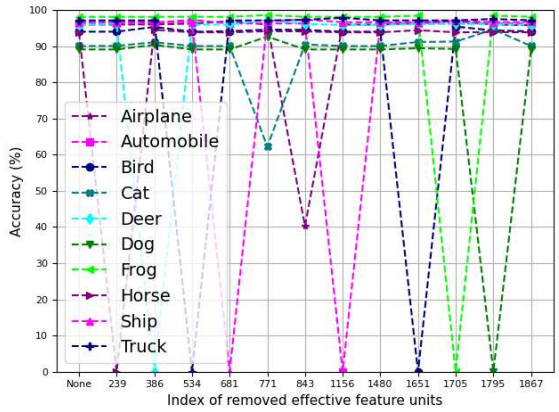
Apart from the proposed MaxAmp-K test, visualization techniques can also be used to interpret the semantic embedding in a representation. To understand whether and how semantic information is captured by individual neural units, we investigate how an individual neural unit at the last layer of the network (i.e., an individual feature, or channel) activates under different inputs. Using a neural network trained under the CIFAR10 dataset, Fig.5 shows how an individual unit/feature responds to an image. For example, the 1588th feature is primarily responsible for the category 'ship', as it outputs a significantly stronger response for the ship category than other categories. This phenomenon can also be seen for other features. These findings show that the compact representation generated by CL-MS can indeed reflect high level patterns or abstract symbols. More importantly, this is achieved without a classifier.

G. Study of network dissection

Using the network dissection tools used in [65]. Fig. 6 shows the classification accuracy when the features/units are removed one by one. Specifically, two neural networks are trained for the CIFAR10 dataset. Out of a total number of 2048 of features, a total of 16 and 8 effective features are identified when the hyper-parameter λ is configured to 5e-5 and 1e-5, respectively. As shown in Fig. 6, the classification accuracy using only these effective features are almost identical to



(a) Default seed



(b) Seed 117

Fig. 7. (a) and (b) represent the accuracy rates of each class for models obtained under different initial conditions after the removal of individual semantic symbols.

results using all the 2048 features. This means classification-related semantic information are fully captured by effective features. When effective features are removed one by one, the classification accuracy reduces gradually. This means most effective features carry certain amount of unique semantic information. We can see that the number of effective features can be controlled by the value of hyper parameter λ. We leave

TABLE V
CLASSIFICATION ACCURACY IN DISTRIBUTED SENSING WITH DIFFERENT DATA FUSION SCHEMES (ACCURACY WITH NODE NUMBERS (1, 2, 3, 4))

Dataset	SNR(dB)	CL(65,536bits)(%)	Max Pro(128bits)(%)	Max Amp(13bits)(%)	Decision(4bits)(%)
CIFAR10	0	**20.32 21.51 23.80 23.95**	10.52 11.21 12.33 13.21	13.12 14.31 14.45 14.49	9.24 9.36 9.59 9.79
CIFAR10	6	**29.73 33.12 39.50 40.15**	16.44 17.35 21.53 22.79	15.66 18.08 20.29 20.35	13.01 13.25 13.39 13.80
CIFAR10	12	**58.58 64.36 65.44 68.49**	45.22 47.16 49.41 51.58	35.21 41.96 44.59 46.94	29.68 29.95 31.00 31.35
CIFAR10	18	**82.42 86.07 86.24 88.80**	81.16 82.42 82.68 84.32	71.44 77.16 79.06 80.23	70.55 70.97 72.77 72.96
CIFAR10	24	**91.55 92.64 93.09 93.28**	91.39 91.44 91.79 91.88	81.66 85.59 87.42 90.00	89.58 90.08 90.36 90.59
		CL(65,536bits)(%)	Max Pro(1536bits)(%)	Max Amp(34bits)(%)	Decision(7bits)(%)
CIFAR100	0	**1.97 2.96 4.20 4.26**	1.32 2.12 2.90 3.47	1.20 1.90 2.50 2.71	0.86 0.92 1.13 1.03
CIFAR100	6	**5.83 8.07 9.13 12.53**	4.51 4.81 6.00 7.27	4.25 4.33 4.57 4.85	1.50 1.52 1.57 1.71
CIFAR100	12	**14.59 17.52 20.10 20.84**	10.90 14.50 17.31 18.84	10.30 12.64 14.16 15.25	5.91 6.13 6.25 6.55
CIFAR100	18	**36.75 40.39 40.63 41.28**	34.69 39.89 40.42 41.26	31.87 35.71 37.67 39.00	25.18 25.63 26.28 26.81
CIFAR100	24	**59.19 61.07 62.12 63.07**	58.57 61.03 61.97 63.02	55.24 58.83 59.85 61.00	53.19 53.29 54.18 54.76

the study of optimally selecting λ for future work.

H. Semantic consistency across different networks

We now perform neural network dissection and show how the proposed methods attain semantic consistency (or invariance). In the same way that different human brains agree on fundamental semantic concepts, different neural network architectures should also agree on the fundamental semantic features of a dataset, i.e., achieve semantic consistency. In Fig. 7, the x-axis shows the indices of (shortlisted effective) neurons that are artificially removed. For example, in Fig. 7(a), 817 on the x-axis means that the 817th neuron is removed from the representation before classification. The result is that the classification accuracy of images of trucks drops to nearly zero. In contrast, the classification accuracy of the images of the other classes remains unchanged. This phenomenon suggests that the 817th neuron in Fig. 7(a) captures the defining features of "trucks". Fig.7(b) shows the same DNN trained with another random seed. The resulting network model and indices of effective neurons are completely different. However, and quite interestingly, one can easily identify the 1651th neuron in Fig. 7(b) as the unique "trucks" feature neuron. We can say that the 817th neuron in Fig. 7(a) and the 1651th neuron in Fig. 7(b) have semantic consistency. Similar degrees of consistency can be easily observed for the other classes.

To quantitatively evaluate the similarity of learned representations using different random seeds, we perform singular value decomposition (SVD) on the index-accuracy matrices whose entries are visualized in Fig. 7 (a) and (b). Specifically, we apply the SVD to decompose the product of the index-accuracy matrix and its transpose. We then calculate the cosine similarity of the principal component vectors obtained from SVD. This allows us to assess the degree of difference between representations trained with different random seeds. For the results shown in Fig. 7, we obtain a cosine similarity of 0.99, indicating a high level of similarity between the two representations learned from different random seeds. This property of *semantic invariance* is a very desirable property for semantic representation learning as it yields a clear interpretation of the representation.

I. Application to distributed sensing

Finally, we consider the potential application of semantic communications to distributed sensing. In our setup, a sensing node captures an independently-distorted version of a source (image in this case), processes and quantizes it to yield binary representation (i.e., bit stream), which is in turn transmitted over lossless communication channels to a Fusion Center (FC) for image classification. The CL-MS scheme is adopted for continuous representation learning. Additive Gaussian noise at different SNR levels is added to the image to simulate distortion.

Four data fusion schemes are considered. The first scheme is decision fusion, in which each sensing node performs classification locally and transmits its decision. A majority-voting mechanism is then applied by the FC to make the final decision. The three remaining schemes consist of feature-based fusion. Among them, one conventional scheme is full feature fusion, which transmits the entire CL-MS continuous representation with double-precision floating point representation. This scheme requires the most bits but achieves the best precision. These schemes represent two extremes. In between we consider the two neural quantization schemes proposed in this paper: MaxPro and MaxAmp quantization. The features received at the FC are added together to train different classifiers when the number of distributed sensing nodes varies from 1 to 4.

Tab. V compares the classification accuracy of the FC output for four data fusion schemes. Experiments are conducted on CIFAR10 and CIFAR100 datasets. We have MaxPro ($N_1(2\sigma) = 16, K = 16, n = 8$) on CIFAR10, MaxPro ($N_1(2\sigma) = 48, K = 48, n = 32$) on CIFAR100, MaxAmp ($N_1(2\sigma) = 16, K = 3, n = 1$) on CIFAR10, and MaxAmp ($N_1(2\sigma) = 48, K = 3, n = 1$) on CIFAR100. It is observed that MaxPro and MaxAmp show progressive performance in terms of compression ratio and classification accuracy between the two extreme: full feature fusion and decision fusion. Overall, at the medium rate region (i.e., around 1 bit per pixel), MaxPro is able to approximate the precision of full feature fusion while achieving better compression ratios by a factor larger than 40. At the extra-low rate region (i.e., around 0.01 bit per pixel), MaxAmp is able to improve the classification accuracy by 12% on average, by almost doubling the number of bits. These results show that the proposed discrete neural representation learning schemes are good candidates for distributed sensing in noisy environments.

TABLE VI
PARAMETERS OF LSTM ARCHITECTURE AND TRAINING PROCEDURE

Parameter Name	Specific Parameter Value
`embedding_dim`	100
`hidden_dim`	256×2
`output_dim`	2
`n_layers`	2
`bidirectional`	`True` (Bidirectional LSTM is used)
`dropout`	0.3
`learning Rate`	0.005
`optimizer`	Adam

TABLE VII
SPARSE METHOD EXTENSION

DataSets	Methods	$N_1(2\sigma)$	$\overline{N_2}$	Accuracy(%)
IMDB	CE	512	512	**88.55**
IMDB	CE-MS($\lambda = 5e-7$)	70	12	83.53
IMDB	CE-MS($\lambda = 1e-7$)	88	14	83.91
IMDB	CE-MS($\lambda = 1e-8$)	98	16	87.05
IMDB	CE-MS($\lambda = 1e-9$)	110	16	<u>87.78</u>
IMDB	CE-FS($\lambda = 5e-7$)	278	58	85.10
IMDB	CE-FS($\lambda = 1e-7$)	354	56	86.36
IMDB	CE-FS($\lambda = 1e-8$)	364	60	87.24
IMDB	CE-FS($\lambda = 1e-9$)	391	68	87.42
IMDB	CE-WS($\lambda = 5e-7$)	70	8	80.26
IMDB	CE-WS($\lambda = 1e-7$)	100	14	83.32
IMDB	CE-WS($\lambda = 1e-8$)	102	16	86.02
IMDB	CE-WS($\lambda = 1e-9$)	112	16	86.30

J. Extension to RNNs

So far, we have limited our investigation to CNNs and image datasets. To further validate the effectiveness of the proposed mask sparsity method to general DNNs, we now consider recurrent neural networks (RNNs). Specifically, we test the task of sentiment analysis (i.e., binary sentiment classification) based on the IMDB dataset of movie reviews [66]. We use a bidirectional long short-term memory (LSTM) network to extract a feature vector representation of size 512. The network parameters are given in Tab. VI. The three sparsity coding methods introduced in Section II.B are applied and tested: mask-based (CE-MS), feature-based (CE-FS), and window-based (CE-WS). For each method, different values of hyper-parameters λ are applied to the feature vector to yield representations with varying degrees of compactness. The resulting accuracy, as well as the two compactness measures ($N_1(2\sigma)$ and $\overline{N_2}$), are compared in Tab. VII. We can see that the proposed CE-MS method generates compact representations, with marginal degradation to the final accuracy compared to the CE (no sparsity) approach. Moreover, CE-MS can simultaneous achieve the best accuracy and compactness performance given by the two conventional sparsity coding approaches (CE-FS and CE-WS). This experiment suggests that the proposed method is a general approach applicable to various types of DNNs.

V. CONCLUSIONS

Semantic representation learning is a fundamental problem in semantic communications. We tackle this problem by proposing a novel DNN design, which aims to learn a compact and semantically meaningful representations. To obtain such representations, we proposed a technique called mask-based sparse coding. Via extensive experiments, we showed that the proposed technique outperforms classic feature-based sparse coding technique by factors ranging from from 9% to 200%, while keeping the same accuracy level, in the tested datasets. To apply mask-based sparse coding for discrete neural representation learning, we have analysed the theoretical bandwidth footprint of our proposed neural index quantization schemes and verified their validity on CIFAR100. We used contrastive learning to encourage semantic alignment and the performance of the resulting scheme was further improved by using sparse coding. We then proposed a test called MaxAmp-K to evaluate the compositionality of the most salient representation output. Experiments using tools of network dissection and visualization showed promising results, suggesting that the proposed sparse contrastive neural networks tend to learn semantic representations with interpretable meanings. Such semantic representations are useful for distributed sensing applications.

VI. FUTURE WORK

In our future work, we will continue to explore various neural network architectures and different data types. In addition, we intend to delve deeper into the automatic adjustment and underlying mechanisms of the hyperparameter λ. Moreover, our research will expand to encompass compression and reconstruction tasks.

ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (Grant No. 62077040, 62125101, and 62341101), the Open Research Fund of the National Mobile Communications Research Laboratory at Southeast University (No. 2023D10), the Natural Science Foundation of Fujian Province of China (No. 2023J01001) the EPSRC's New Investigator Award (EP/T026111/1) and the New Cornerstone Science Foundation through the XPLORER PRIZE.

REFERENCES

[1] F. Yang, S. Zhu, H. Li, X. Huang, Y. Sun, and J. Tian, "A novel architecture design of power internet of things based on huawei internet of things platform," in *Proc. Conf. IEEE Ene. Int.(EI2)*, Nov. 2022, pp. 1875–1881.

[2] F. Jameel, Z. Chang, J. Huang, and T. Ristaniemi, "Internet of autonomous vehicles: Architecture, features, and socio-technological challenges," *IEEE Wirel. Commun.*, vol. 26, no. 4, pp. 21–29, Aug. 2019.

[3] E. C. Strinati and S. Barbarossa, "6G networks: Beyond shannon towards semantic and goal-oriented communications," *Comput. Netw.*, vol. 190, no. 107930, May. 2021.

[4] H. Xie, Z. Qin, G. Y. Li, and B.-H. Juang, "Deep learning enabled semantic communication systems," *IEEE Trans. Signal Process.*, vol. 69, pp. 2663–2675, Apr. 2021.

[5] P. Zhang, W. Xu, H. Gao, K. Niu, X. Xu, X. Qin, C. Yuan, Z. Qin, H. Zhao, J. Wei *et al.*, "Toward wisdom-evolutionary and primitive-concise 6G: A new paradigm of semantic communication networks," *Engineering*, vol. 8, pp. 60–73, Jan. 2022.

[6] X. Luo, H.-H. Chen, and Q. Guo, "Semantic communications: Overview, open issues, and future research directions," *IEEE Wirel. Commun.*, vol. 29, no. 1, pp. 210–219, Jan. 2022.

[7] G. Shi, D. Gao, X. Song, J. Chai, M. Yang, X. Xie, L. Li, and X. Li, "A new communication paradigm: From bit accuracy to semantic fidelity," *arXiv:2101.12649*, 2021. [Online]. Available: https://arxiv.org/abs/2101.12649

[8] H. Xie, Z. Qin, and G. Y. Li, "Task-oriented multi-user semantic communications for VQA task," *IEEE Wirel. Commun. Lett.*, vol. 11, no. 3, pp. 553–557, Dec. 2022.

[9] Q. Zhou, R. Li, Z. Zhao, C. Peng, and H. Zhang, "Semantic communication with adaptive universal transformer," *IEEE Wirel. Commun. Lett.*, vol. 11, no. 3, pp. 453–457, Dec. 2022.

[10] M. Kountouris and N. Pappas, "Semantics-empowered communication for networked intelligent systems," *IEEE Commun. Mag.*, vol. 59, no. 6, pp. 96–102, Jun. 2021.

[11] H. Xie and Z. Qin, "A lite distributed semantic communication system for internet of things," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 142–153, Nov. 2020.

[12] M. Sana and E. C. Strinati, "Learning semantics: An opportunity for effective 6G communications," in *Proc. IEEE 19th Annu. Consum. Commun. Netw. Conf. (CCNC)*, Las Vegas, NV, USA, Jan. 2022, pp. 631–636.

[13] M. K. Farshbafan, W. Saad, and M. Debbah, "Curriculum learning for goal-oriented semantic communications with a common language," *arXiv:2204.10429*, 2022. [Online]. Available: https://arxiv.org/abs/2204.10429

[14] H. Seo, J. Park, M. Bennis, and M. Debbah, "Semantics-native communication with contextual reasoning," *arXiv:2108.05681*, 2021. [Online]. Available: https://arxiv.org/abs/2108.05681

[15] Z. Lei, P. Duan, X. Hong, J. F. C. M. Mota, J. Shi, and C.-X. Wang, "Progressive deep image compression for hybrid contexts of image classification and reconstruction," *IEEE J. Sel. Areas Commun*, vol. 41, no. 1, pp. 72–89, Dec. 2023.

[16] Y. Fu, S. Wang, C.-X. Wang, X. Hong, and S. McLaughlin, "Artificial intelligence to manage network traffic of 5g wireless networks," *IEEE Network*, vol. 32, no. 6, pp. 58–64, Nov. 2018.

[17] S. Harnad, "The symbol grounding problem," *Physica D*, vol. 42, no. 1-3, pp. 335–346, Jun. 1990.

[18] S. Tellex, T. Kollar, S. Dickerson, M. R. Walter, A. G. Banerjee, S. Teller, and N. Roy, "Approaching the symbol grounding problem with probabilistic graphical models," *AI Mag.*, vol. 32, no. 4, pp. 64–76, Dec. 2011.

[19] Y. Bar-Hillel and R. Carnap, "Semantic information," *Br. J. Philos. Sci.*, vol. 4, no. 14, pp. 147–157, Aug. 1953.

[20] Carnap, Rudolf, and Y. Bar-Hillel, "An outline of a theory of semantic information," *British Journal for the Philosophy of Science*, vol. 4, Jun. 1953.

[21] L. Floridi, "Outline of a theory of strongly semantic information," *Minds Mach.*, vol. 14, no. 2, pp. 197–221, Mar. 2004.

[22] F. Willems and T. Kalker, "Semantic compaction, transmission, and compression codes," in *Proc. Int. Symp. Inf. Theory (ISIT)*, Adelaide, SA, Australia, Sep. 2005, pp. 214–218.

[23] J. Bao, P. Basu, M. Dean, C. Partridge, A. Swami, W. Leland, and J. A. Hendler, "Towards a theory of semantic communication," in *Proc. Workshop Net. Sci(NS).*, NSW, Australia, Jun. 2011, pp. 110–117.

[24] B. Güler, A. Yener, and A. Swami, "The semantic communication game," *IEEE Trans. Cogn. Commun. Netw.*, vol. 4, no. 4, pp. 787–802, Sep. 2018.

[25] A. Shoshan, N. Bhonker, I. Kviatkovsky, and G. Medioni, "Gan-control: Explicitly controllable gans," in *Proc. Int. Conf. Comput. Vis.(ICCV)*, Montreal, BC, Canada., Oct. 2021, pp. 14 083–14 093.

[26] W. Ahmed, P. Morerio, and V. Murino, "Cleaning noisy labels by negative ensemble learning for source-free unsupervised domain adaptation," in *Proc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, New Orleans, Louisiana., Jan. 2022, pp. 1616–1625.

[27] Y. Tian, D. Krishnan, and P. Isola, "Contrastive multiview coding," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Glasgow, US, Jul. 2020, pp. 776–794.

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

[29] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," in *Pro. Adv. Neural Inf. Process. Syst. (NIPS)*, Vancouver, Canada, Dec. 2020, pp. 18 661–18 673.

[30] T. Wang and P. Isola, "Understanding contrastive representation learning through alignment and uniformity on the hypersphere," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Vienna, Austria, Jun. 2020, pp. 9929–9939.

[31] M. Garnelo and M. Shanahan, "Reconciling deep learning with symbolic artificial intelligence: Representing objects and relations," *Curr. Opin. Behav. Sci.*, vol. 29, pp. 17–23, Oct. 2019.

[32] A. Mitrokhin, P. Sutor, D. Summers-Stay, C. Fermuller, and Y. Aloimonos, "Symbolic representation and learning with hyperdimensional computing," *Frontiers in Robotics and AI*, vol. 7, p. 63, Jun. 2021.

[33] J. Ascenso, E. Alshina, and T. Ebrahimi, "The jpeg ai standard: Providing efficient human and machine visual data consumption," *Ieee Multimedia*, vol. 30, no. 1, pp. 100–111, 2023.

[34] W. Gao, S. Liu, X. Xu, M. Rafie, Y. Zhang, and I. Curcio, "Recent standard development activities on video coding for machines," *arXiv preprint arXiv:2105.12653*, 2021.

[35] A. Van Den Oord, O. Vinyals *et al.*, "Neural discrete representation learning," *Advances in neural information processing systems*, vol. 30, 2017.

[36] Y. Lemesle, M. Sawayama, G. Valle-Perez, M. Adolphe, H. Sauzéon, and P.-Y. Oudeyer, "Language-biased image classification: evaluation based on semantic representations," in *Proc. Workshop Int. Conf. Learn. Representations. (ICLR)*, Virtual conference, May. 2021.

[37] C. M. Rodrigues, L. Pereira, A. Rocha, and Z. Dias, "Image semantic representation for event understanding," in *Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS)*. Delft, The Netherlands: IEEE, Dec. 2019, pp. 1–6.

[38] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *Proc. Int. Conf. Learning Representations. (ICLR)*, Banff, Canada, Apr. 2014.

[39] L. Meng, S. Ding, and Y. Xue, "Research on denoising sparse autoencoder," *Int. J. Mach. Learn. Cybern.*, vol. 8, no. 5, pp. 1719–1729, May. 2017.

[40] E. J. Candes, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted l1 minimization," *J. Fourier Anal*, vol. 14, pp. 877–905, Oct. 2008.

[41] M. A. Khajehnejad, W. Xu, A. S. Avestimehr, and B. Hassibi, "Analyzing weighted ℓ_1 minimization for sparse recovery with nonuniform sparse models," *IEEE Trans. Signal Process*, vol. 59, no. 5, pp. 1985–2001, Jan. 2011.

[42] J. F. Mota, L. Weizman, N. Deligiannis, Y. C. Eldar, and M. R. Rodrigues, "Reference-based compressed sensing: A sample complexity approach," in *IEEE Conf. Acoustics, Speech and Signal Proc. (ICASSP)*. IEEE, 2016, pp. 4687–4691.

[43] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Munich, Germany, Sep. 2018, pp. 8–14.

[44] J. Park, S. Woo, J.-Y. Lee, and I. S. Kweon, "BAM: Bottleneck attention module," *arXiv:1807.06514*, 2018. [Online]. Available: https://arxiv.org/abs/1807.06514

[45] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.

[46] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Workshop Int. Conf. Learn. Representations. (ICLR)*, San Diego, CA, USA, May. 2015.

[47] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778.

[48] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.(NIPS)*, vol. 30, Long Beach, USA, Dec. 2017.

[49] R. Krishnamoorthi, "Quantizing deep convolutional networks for efficient inference: A whitepaper," *arXiv:1806.08342*, 2018. [Online]. Available: https://arxiv.org/abs/1806.08342

[50] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks," in *Proc. Adv. Neural Inf. Process. Syst.(NIPS)*. Bacelona, Spain: Curran Associates, Inc., Dec. 2016.

[51] J. Chen, Y. Liu, H. Zhang, S. Hou, and J. Yang, "Propagating asymptotic-estimated gradients for low bitwidth quantized neural networks," *IEEE J. Sel. Top. Signal Process.*, vol. 14, no. 4, pp. 848–859, Jan. 2020.

[52] A. Razavi, A. van den Oord, and O. Vinyals, "Generating diverse high-fidelity images with VQ-VAE-2," in *Proc. Adv. Neural Inf. Process. Syst.(NIPS)*. Vancouver, Canada: Curran Associates, Inc., Dec. 2019.

[53] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, "Zero-shot text-to-image generation," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Vienna, Austria, Jul. 2021, pp. 8821–8831.

[54] Y. Zhang, P. Tio, A. Leonardis, and K. Tang, "A survey on neural network interpretability," *IEEE Trans. Emerging Top. Comput. Intell.*, vol. 5, no. 5, pp. 726–742, 2021.

[55] W. Maass, "Networks of spiking neurons: the third generation of neural network models," *Neural networks*, vol. 10, no. 9, pp. 1659–1671, 1997.

[56] A. Mostaani, T. X. Vu, S. Chatzinotas, and B. Ottersten, "Task-oriented data compression for multi-agent communications over bit-budgeted channels," *IEEE Open J. Commun. S.*, vol. 3, pp. 1867–1886, 2022.

[57] ——, "Task-effective compression of observations for the centralized control of a multi-agent system over bit-budgeted channels," *arXiv preprint arXiv:2301.01628*, 2023.

[58] A. Krizhevsky, "Learning multiple layers of features from tiny images," *University of Toronto*, May. 2012.

[59] G. Luo, "A review of automatic selection methods for machine learning algorithms and hyper-parameter values," *Network Model. Anal. Health Inf. Bioinf.*, vol. 5, pp. 1–16, 2016.

[60] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Proc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Salt Lake City, UT, USA, Jun. 2018, pp. 7482–7491.

[61] P. Stoica and A. B. Gershman, "Maximum-likelihood doa estimation by data-supported grid search," *IEEE Signal Process Lett.*, vol. 6, no. 10, pp. 273–275, 1999.

[62] H. Ouda, A. Badr, A. Rashwan, H. S. Hassanein, and K. Elgazzar, "Optimizing real-time ecg data transmission in constrained environments," in *Proc. IEEE Int'l. Conf. Common(ICC)*, Seoul, Korea, May. 2022, pp. 2114–2119.

[63] R. C. Daniels and S. W. Peters, "A new mimo hf data link: Designing for high data rates and backwards compatibility," in *Proc. Conf. IEEE Mil. Commun. (MILCOM)*. San Diego, California, USA.: IEEE, Nov. 2013, pp. 1256–1261.

[64] P. Majumder, K. Sinha, and B. P. Sinha, "Dcv ns: A new energy efficient transmission scheme for wireless sensor networks," in *Proc. IEEE Conf. Veh. Tech. Common(VTC)*, Chicago, IL, USA, Aug. 2018, pp. 1–5.

[65] D. Bau, J.-Y. Zhu, H. Strobelt, A. Lapedriza, B. Zhou, and A. Torralba, "Understanding the role of individual units in a deep neural network," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 117, no. 48, pp. 30 071–30 078, Sep. 2020.

[66] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proc. Annu. Meet. Assoc. Comput. Linguist.: Hum. Lang. Technol.(ACL-HLT)*. Portland, Oregon, USA: Association for Computational Linguistics, Jun. 2011, pp. 142–150. [Online]. Available: http://www.aclweb.org/anthology/P11-1015

Wenhui Hua (Student Member, IEEE) received the B.S. degree from Fuzhou University. He is currently pursuing the Ph.D. degree at National-Local Joint Engineering Research Center of Navigation and Location Services, Xiamen University, Xiamen, China. His current research interests include semantic communication and symbolic representation learning.

Longhui Xiong received the B.S. degree from Southeast University. He is currently pursuing the M.S. degree at School of Informatics, Xiamen University, Xiamen,China. His current research interests include semantic communication and deep learning.

Sicong Liu (Senior Member, IEEE) received the B.S.E. and Ph.D. degrees (Highest Hons.) in electronic engineering from Tsinghua University, Beijing, China, in 2012 and 2017, respectively. He is an Associate Professor with the Department of Information and Communication Engineering, School of Informatics, Xiamen University, Xiamen, China. He was a Senior Engineer with Huawei Technologies Company Ltd., China, from 2017 to 2018. He was a Visiting Scholar with the City University of Hong Kong in 2010. His current research interests are compressed sensing, AI-assisted communications, integrated sensing and communications, and visible light communications. He has authored over 60 journal or conference papers, and four monographs in the related areas. Dr. Liu won the Best Paper Award at ACM UbiComp 2021 CPD WS as the corresponding author, and the Second Prize in the Natural Science Award of Chinese Institute of Electronics. He has served as the associate editor or TPC chair of several IEEE and other international academic journals and conferences. He is a Senior Member of China Institute of Communications.

Lingyu Chen (Member, IEEE) received his B.S. degree in software engineering (2006) and the Ph.D. degree in communication and information systems (2011), both from Xiamen University. He is currently an associate Professor in the School of Information Science and Technology, Xiamen University, China. His research interests include wireless sensor networks, acoustic sensor networks, edge computing, semantic communication and wireless signal processing.

Xuemin Hong (Member, IEEE) received the Ph.D. degree from Heriot-Watt University, Edinburgh, U.K., in 2008. He is currently a Professor with the School of Informatics, Xiamen University, China. He has published over 60 articles in refereed journals and conference proceedings. His current research interests include semantic communications, cognitive communication networks, and wireless localization systems.

João F. C. Mota (Member, IEEE) received the M.Sc. and Ph.D. degrees in electrical and computer engineering from the Technical University of Lisbon in 2008 and 2013, respectively, and the Ph.D. degree in electrical and computer engineering from Carnegie Mellon University in 2013. He is currently an Assistant Professor of signal and image processing with Heriot-Watt University, Edinburgh. His research interests include theoretical and practical aspects of high-dimensional data processing, inverse problems, optimization theory, machine learning, data science, and distributed information processing and control. He was a recipient of the 2015 IEEE Signal Processing Society Young Author Best Paper Award.

Xiang Cheng (Fellow, IEEE) received the Ph.D. degree jointly from Heriot-Watt University and the University of Edinburgh, Edinburgh, U.K., in 2009. He is currently a Boya Distinguished Professor of Peking University. His general research interests are in areas of channel modeling, wireless communications, and data analytics, subject on which he has published more than 280 journal and conference papers, 9 books, and holds 17 patents. Prof. Cheng is a Distinguished Young Investigator of China Frontiers of Engineering, a recipient of the IEEE Asia Pacific Outstanding Young Researcher Award in 2015, a Distinguished Lecturer of IEEE Vehicular Technology Society, and a Highly Cited Chinese Researcher in 2020. He was a co-recipient of the 2016 IEEE JSAC Best Paper Award: Leonard G. Abraham Prize, and IET Communications Best Paper Award: Premium Award. He has also received the Best Paper Awards at IEEE ITST'12, ICCC'13, ITSC'14, ICC'16, ICNC'17, GLOBECOM'18, ICCS'18, and ICC'19. He has served as the symposium lead chair, co-chair, and member of the Technical Program Committee for several international conferences. He is currently a Subject Editor of IET Communications and an Associate Editor of the IEEE Transactions on Wireless Communications, IEEE Transactions on Intelligent Transportation Systems, IEEE Wireless Communications Letters, and the Journal of Communications and Information Networks. In 2021, he was selected into two world scientist lists, including World's Top 2% Scientists released by Stanford University and Top Computer Science Scientists released by Guide2Research.