Discrete Messages Improve Communication Efficiency among Isolated Intelligent Agents

Hang Chen[1*], Yuchuan Jiang[1*], Weijie Zhou[2], Cristian meo[3], Ziwei Chen[1], Dianbo Liu[4]

[1]BeiJing JiaoTong University.
[2]Rice University.
[3]Delft University of Technology.
[4]School of Medicine and College of Design&Engineering, National University of Singapore.

Contributing authors: zwchen@bjtu.edu.cn or dianbo@nus.edu.sg;

Abstract

Individuals, despite having varied life experiences and learning processes, can communicate effectively through languages. This study aims to explore the efficiency of language as a communication medium. We put forth two specific hypotheses: First, discrete messages are more effective than continuous ones when agents have diverse personal experiences. Second, communications using multiple discrete tokens are more advantageous than those using a single token. To validate these hypotheses, we designed multi-agent machine learning experiments to assess communication efficiency using various information transmission methods between speakers and listeners. Our empirical findings indicate that, in scenarios where agents are exposed to different data, communicating through sentences composed of discrete tokens offers the best inter-agent communication efficiency. The limitations of our finding include lack of systematic advantages over other more sophisticated encoder-decoder model such as variational autoencoder and lack of evluation on non-image dataset, which we will leave for future studies.

Keywords: discrete messages, multiple discrete tokens, inter-agent communication

1 Introduction

Intelligent agent communication, situated at the crossroads of AI and linguistics, aims to explore how a common language develops among agents. The Lewis Game[20] exemplifies collaborative tasks in this field, where a speaker and a listener collaborate to accomplish a task. The speaker describes a specific object to the listener, who must correctly identify it from a set of alternatives (Figure 1(left)). Using AI, significant research[31][16][35] has investigated language origins and evolution. Chaabouni et al[4] conducted research on emergent communication[15], focusing on various aspects such as the assessment of compositionality and generalization in emerging language agents and the development of efficient color naming systems[5]. Lazaridou et al[19] proposed a method that combines multi-agent communication with data-driven natural language learning, aiming to enable machine agents to effectively communicate with humans using natural language. Lowe et al[23] investigated the interaction between supervised learning and self-play in protocol learning for emergent communication. Deep reinforcement learning has also been applied in the field of multi-agent communication[30][7]. However, as the number of agents increases, redundant and indistinguishable information often leads to inefficient communication.Yet, with more agents comes the challenge of redundancy and inefficiency. Recent work[26][28][6][3] has delved into optimizing communication, surpassing simple capacity and extending dataset complexity, significantly contributing to our understanding of language's evolution.

Vector Quantization (VQ) adheres to Shannon's rate-distortion theory[9], suggesting that vector encoding may outperform scalar encoding by handling dependencies in source symbols adequately. Advances in reparameterization, specifically for VAEs handling discrete variables[29][24], have improved the model's effectiveness. VQVAE[33] circumvents non-differentiability issues by applying the identity function for efficient gradient transmission. This vector-based discrete representation is shown to be more robust and generalizable than its continuous counterparts for complex learning models[33][22]. Furthermore, applying discretization in multi-agent reinforcement learning addresses communication challenges within modular reasoning architectures, enhancing efficient interactions across modules[11].

Building upon the prior research, this study presents advancements in the vector quantization technique of the VQ model. We have made the following hypotheses in this paper: When intelligent agents have diverse personal experiences, using discrete messages is more advantageous than continuous messages, and communicating using multiple discrete tokens is more advantageous than using a single token. Through our designed experiments in multi-agent machine learning, we have empirically demonstrated that communication using sentences composed of multiple discrete tokens offers superior communication efficiency among agents with diverse personal experiences. However, in the experiments, when we use the VAE model to simulate continuous language communication between agents instead of the AE model, its effectiveness is superior to that of communication using multi-token discrete language.

The structure of our paper is as follows: Section 1 provides the background of emergent communication and outlines the objectives of our research. In Section 2, we introduce the related work in the field. Section 3 describes the experimental settings

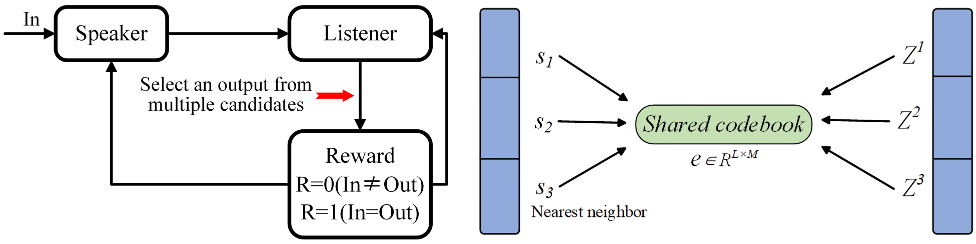

Fig. 1 Left: Lewis Game. Right: In structured architectures, communication is typically vectorized. The communication vector is initially divided into multiple discretization tokens. Each token goes through separate discretization, where it is quantized to the nearest neighbor within a shared collection of latent codebook vectors. Subsequently, the discretization tokens are concatenated back together to form a vector with the same shape as the original one.

and methods employed in our study. In Section 4, we present the experimental results and draw meaningful conclusions. Finally, in Section 5, we conclude our research with discussions and prospects for future work.

2 Related Works

In recent years, numerous approaches have emerged to facilitate effective communication among specialized components within machine learning models. These approaches include attention mechanisms that selectively transmit information between specialized components in machine learning models[11][2][10] and the Transform method[34][18]. Additionally, collective memory and shared parameter techniques have been utilized for communication in multi-agent settings[27].

The RIAL model is widely used for discrete communication in intelligent agent coordination, enabling communication through discrete symbols, mirroring human social interaction[7]. Guo et al.[12] explored the importance of computer simulations in evolutionary linguistics using intelligent agent models to simulate agents developing a compositional language for numerical concepts through communication. Studies into LSTM[14] language models by Lakretz et al.[17] have shed light on how hidden states represent numbers and syntactic structures, guiding further inquiry into language patterns. Further work by Miao et al.[25], Garcia et al.[8], and Havrylov et al.[13] expanded our understanding of multi-agent communication dynamics and the emergence of language in neural network-based agents.

In this vein, our research employs the Vector-Quantized Variational Autoencoder (VQVAE) model[33] and utilizes cross-training and cross-validation to probe communication patterns amongst agents. Our experiments demonstrate that in settings where agents have varied language systems, discrete language proves more effective than continuous forms. We also examine how differing token quantities in codebooks influence discrete communication effectiveness, a topic which is further elaborated in Section 4.

3 Theoretical Basis and Experimental Method

3.1 Discrete and Continuous Communication

In the series of autoencoder models[1], we designate the encoder component of the model as the speaker, while the decoder component is referred to as the listener.

For the discrete communication model, we use VQVAE. To train a pair of agents, let's assume the input is x. The information is passed through the speaker as $z_e(x) = e(x, \theta)$, representing the encoded representation of x. Then, the information undergoes discrete quantization using a codebook, resulting in $Z = DISCRETIZE(z_e(x), \varphi)$. Finally, the speaker reconstructs the original information from the received codebook indices $x^{'} = d(Z, \phi)$, where the model parameters θ, φ, ϕ are continuously updated by minimizing the reconstruction loss and codebook loss. The complete loss function for this process is as shown in Equation 1:

$$\mathbb{L}_{VQ} = ||x - x^{'}||_2 + ||sg[z_e(x)] - e_k||_2^2 + \beta||z_e(x) - sg[e_k]||_2^2 \tag{1}$$

Where the last two terms represent the codebook loss in the VQVAE model, in the subsequent algorithm, we use $L_{quantify}$ to represent these two items. The continuous data output by the encoder is quantized by the codebook layer before being transmitted to the decoder, which is the process of discrete communication between agents as we define it. In the experiments involving the AE model, the overall loss can be expressed as:$z = e(x, \theta)$,$x^{'} = d(z, \phi)$. As shown in Equation 2, the overall loss is equivalent to the reconstruction loss.

$$\mathbb{L}_{AE} = ||x - x^{'}||_2 = ||x - d(e(x, \theta), \phi)||_2 \tag{2}$$

During this process, the continuous data output by the encoder is directly input into the decoder, which is the process of agents using continuous language for communication.

Throughout the entire experiment, the experimental data based on the Autoencoder (AE) serves as a baseline, which aims to verify that under the same experimental settings, the use of continuous language communication is less effective than discrete semantic communication between unfamiliar agents.

3.2 Multi-token Discretization

Li et al[21] proposed a human-like discrete information generation method that enables discrete message communication to have the effect of continuous message communication. Based on the foundation of discretization, we propose a multi-token discretization approach. In the original autoencoder (AE) framework, input data is encoded by the encoder into a continuous vector. The VQVAE model builds upon the AE and introduces a latent space codebook between the encoder and decoder. The continuous variables from the encoder are quantized into multiple vectors of the same size as the codebook. In our research, multi-token discretization is applied before the data enters the codebook layer. It involves dividing the output of the encoder into multiple segments of equal size but containing different data. Let's assume our latent codebook size is $e \in R^{L \times M}$. Initially, the output $z_e(x)$ is divided into N segments $s_1, s_2, s_3, ...s_N$ with $z_e(x) = CONCAT(s_1, s_2, s_3, ...s_N)$, where each segment $s_i \in \mathbb{R}^{\frac{M}{N}}$ with $\frac{M}{N} \in N^+$. Next, each of these segments is discretized sequentially: $e_{o_i} = Discretize(s_i)$, where $o_i = argmin||s_i - s_j||(j \in 1, ...N)$. After the discretization process, the N segments of data are then integrated back together in the order of their original splitting, as shown in Equation 3:

$$Z = CONCAT(Discretize(s_1), Discretize(s_2), ...Discretize(s_N)) \tag{3}$$

Throughout the entire process, the discretized multi-token data always shares the same codebook. The schematic diagram of the multi-token discretization is illustrated in Figure 1(Right). After modifying this part of the structure, the corresponding model loss function needs to be adjusted as well. Since we divide the data into N segments, the total loss function for model training is defined as shown in Equation 4.

$$\mathbb{L} = \mathbb{L}_{task} + \frac{1}{N} \left\{ \sum_{i=1}^{N} ||sg[s_i] - e_{o_i}||_2^2 + \beta \sum_{i=1}^{N} ||s_i - sg[e_{o_i}]||_2^2 \right\} \tag{4}$$

Where L_{task} represents the specific task loss, which can be the aforementioned reconstruction loss, classification loss, or any other relevant loss function.

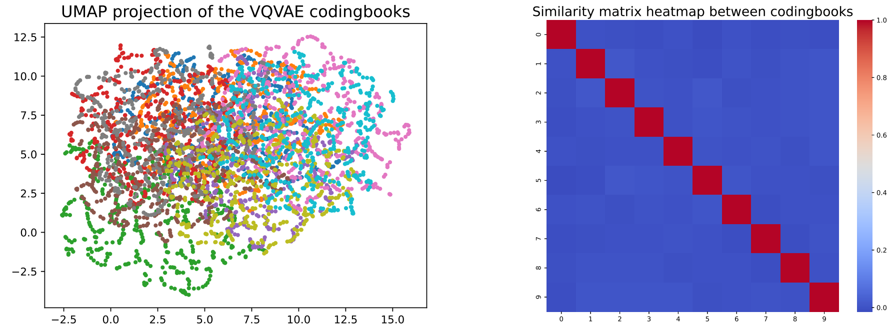

Fig. 2 Ten pairs of agents' understanding of the same language. Left: Feature distribution of latent codebooks for different agents. Right: Similarity of different latent codebooks. The understanding of this language is different for each pair of agents.

3.3 Learning and Validation of Communication for Agents

Attempts have been made to explore cross-training (Guo et al[12], Tieleman et al[32]) in the context of multi-agent learning. In this approach, during the simultaneous training of multiple agents, after each iteration, a random combination is selected, pairing one agent's speaker with another agent's listener for the next round of iterative learning. The reason behind this approach is that when multiple agents learn the same language, their understanding of the language may not be entirely identical. Figure 2 demonstrate the feature distributions of the latent codebook spaces for 10 pairs of agents trained simultaneously on the same MNIST dataset. Each color represents a communication protocol between a pair of agents, that is, the feature distribution in the codebook. It can be observed that there are differences in semantic understanding among the agents. Hence, cross-training becomes necessary because it allows different agents to have the most similar understanding of the same language. Algorithm 2 in the Appendix implements the aforementioned process.

Algorithm 1 Individual Training with Cross-Validation

Require: Using the processed dataset: $Dataset^{'} = (dataset^{'}_1, dataset^{'}_2, ...dataset^{'}_m)$;
Ensure: Train m pairs of agents simultaneously.
Initialize encoders $E = \{e_0, ..., e_m\}$.
Initialize codebooks $C = \{c_0, ..., c_m\}$.
Initialize decoders $D = \{d_0, ..., d_m\}$.
for each iteration i **do**
  for each pair of agents j **do**
    Sample input data x_j from $dataset^{'}_j$
    $L_j \Leftarrow d_j(c_j(e_j(x_j)))$
    optimize e_j,c_j and d_j with respect to L_j
  end for
end for
Validation:
for each pair of agents j **do**
  Sample validate data x
  $loss_j \Leftarrow d_k(c_j(e_j(x))), (k = 1, 2...m, k \neq j)$
  output $loss_j$
end for

In terms of experimental validation, in addition to using trained agents for verification, we conducted another form of validation by manipulating the dataset. Assuming there are m types of data in the dataset, we merged the training set and validation set into a single dataset. The merged dataset was then divided into m classes based on their labels $Dataset = (dataset_1, dataset_2, ...dataset_m)$. After the division, a portion of images was uniformly sampled from each class to form the validation set, denoted as $Validataset = (sample^1_1, sample^1_2, ...sample^1_m)$. To meet the experimental overlap requirements, from the remaining training set P_{train}, a certain number of images were extracted from each class according to the desired experimental overlap rate across the classes $Overlapset = (sample^2_1, sample^2_2, ...sample^2_m)$. There are an equal number of images in each class, and P_{train} is the number of images left after the first extraction. For $p_j \in \{0.05, 0.1, 0.2, ...0.9\}$, the calculation of the number of images sampled in the second extraction is given by Equation 5.

$$\frac{m * sample^2_i}{P_{train} + (m-1) * sample^2_i} = p_j, (i = 1, 2..., m) \tag{5}$$

These extracted images were then merged with the respective training sets, ensuring that each class in the training set contained images from the remaining $m-1$ classes.

The processed training set is $Dataset' = (dataset'_1, dataset'_2, ...dataset'_m)$. This process resulted in a training set where each category served as a separate training set for a pair of agents to learn from. The experiments conducted on the split dataset follow Algorithm 1, which forms the core of our research paper. Specifically, discrete messages are more effective than continuous ones when agents have diverse personal experiences. Similar to Algorithm 1, the experimental methodology of our core content is illustrated in Figure 3.

The aforementioned are the two experimental procedures we used to explore the communication patterns of multiple agents. The main model involved in the procedures is the VQVAE model. However, when we incorporate the AE (Autoencoder) model in our experiments, we simply remove the ”c” module from the procedure, and the data outputted by the encoder is directly decoded by the decoder.

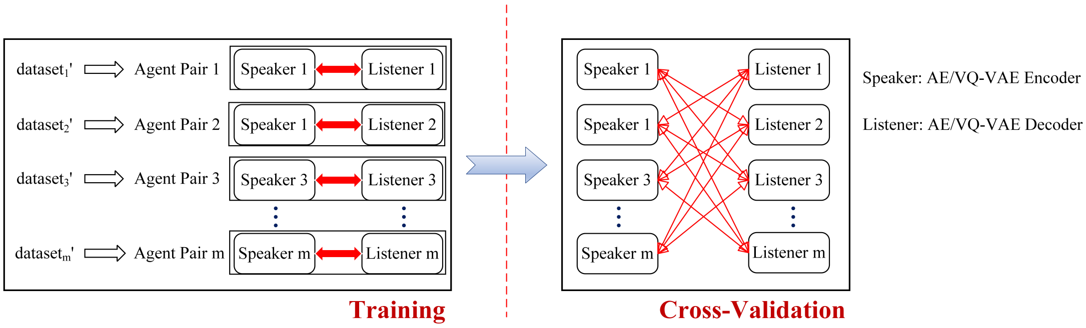

Fig. 3 Training and validation of paired agents. Each pair of agents has its own dataset during training and learning. Upon completion of learning, one agent from the pair interacts with the other agent from another pair. This is our core methodology, where in this validation scenario, the advantage of discrete language in communication between agents is determined based on the reconstruction losses of information.

4 Experiments

The models used in this paper were implemented on the PyTorch 1.12.1 framework, using PyCharm Community Edition 2023.1 on the Windows platform. The model training was conducted on a single GeForce RTX 3060 GPU with 8GB of GPU memory, using the CUDA 12.1 experimental environment. The operating system used was Windows. In our work, we employed three datasets: MNIST, CIFAR10, CelebA and our own medical dataset. The image resolution for all three datasets was separately set to $28 \times 28, 32 \times 32, 64 \times 64$ and 64×64. The batch size for the first two datasets during training is set to 256, while the batch size for the latter two datasets is set to 64, and we utilized the Adam optimizer with a learning rate of 0.001. The commitment cost for the model's discrete layer was set to 0.25, with a decay rate of 0.99. The specific codebook size for the discrete layer varied depending on the dataset. In the experiments, we compared the reconstruction errors between the original images and the reconstructed images. An example of the two types of images can be seen in Figure 4.

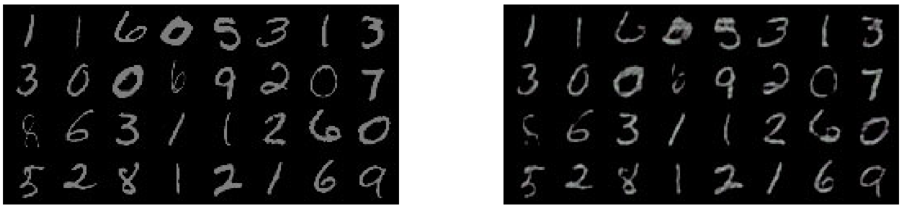

Fig. 4 Example images for the reconstruction task. Left: Original image. Right: Reconstructed image.

4.1 Multi-token Discretization for Improved Agent Communication

According to the method shown in Figure 3 and Algorithm 1, we repeated the experiments with different overlap ratios using the multi-token discretization model with the best performance and the AE model. For the MNIST and CelebA datasets, the original VQVAE model had latent space size $e_m \in \mathbb{R}^{512\times 64}$ and $e_m \in \mathbb{R}^{512\times 128}$, while for the CIFAR10 dataset, it was $e_m \in \mathbb{R}^{1024\times 256}$. We conducted the above overlap experiments with 32 tokens, and the experimental results are shown in Figure 5. Under three different datasets, the average loss incurred by using multiple discrete tokens for communication is 32.1%, 10.6%, and 3.7% lower than that incurred by using continuous semantics for communication, respectively. Our experiments indicate that when one agent interacts with another unfamiliar agent, the discrete semantic learning method using multiple tokens has certain advantages over continuous semantic learning. Figure

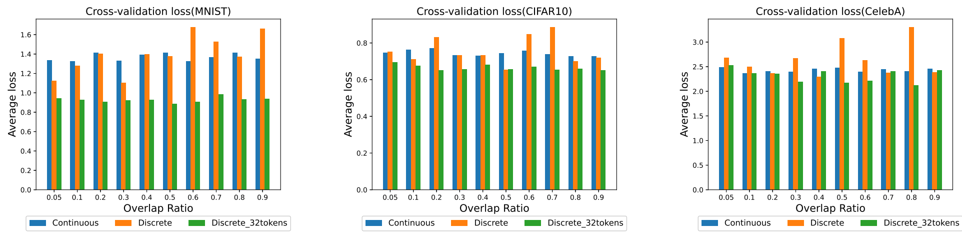

Fig. 5 Communication test loss on three types of models. The bar graphs above represent the losses for m pairs of agents interacting with continuous variables, discrete variables, and multi-token discrete variables under different dataset overlap probabilities. The communication between agents that learn and interact with multi-token discrete variables is the most effective.

6 explain why we chose to conduct our experiments with a 32-token VQVAE model as mentioned above and also illustrate the advantages of our proposed multi-token discrete mechanism compared to a single-token approach. It shows the results of training m pairs of agents simultaneously according to Algorithm 1, where each boxplot in the figure represents the stable loss from communications between the m pairs of agents. The general pattern is that as the number of discrete tokens increases, the communication loss decreases, which from the model's perspective, leads to improved performance, and in terms of agent communication, this makes their communication more efficient.

In all of the above experiments, the number of agent pairs m for the three datasets respectively are $(m = 10(MNIST, CIFAR10), 8(CelebA))$. Our experiments have demonstrated two theories. First, discrete messages with multiple tokens are more

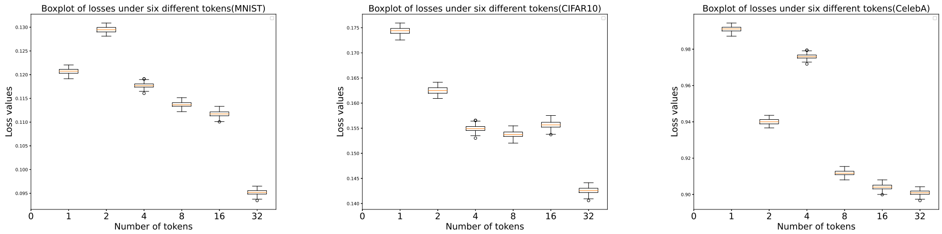

Fig. 6 The communication loss of multiple agents under multi-token discretization. The increase in the number of tokens can reduce the loss from communications. Left: MNIST, codebook size $e_m \in \mathbb{R}^{512\times64}$. Mid: CIFAR10, codebook size $e_m \in \mathbb{R}^{1024\times256}$. Right: CelebA, codebook size $e_m \in \mathbb{R}^{512\times64}$.

effective than continuous ones when agents have diverse personal experiences. Second, communications using multiple discrete tokens are more advantageous than those using a single token.

4.2 Theoretical Validation and Practical Application

Based on the open-source datasets, we conducted the same experiments as in section 4.1 with our own ocular dataset(Figure 7) to further validate our theory. When experimenting with the VQVAE model on this dataset, codebook size $e_m \in \mathbb{R}^{512\times128}$. The dataset is divided into 5 categories based on symptom types, with varying numbers of images in each category. We performed 5 sets of experiments, each involving communications between agents. First, to demonstrate the feasibility of our proposed multi-token discretization, we conducted experiments following Algorithm 2. The experimental results are shown in Figure 8(Right), which once again validate the effectiveness of our theory.

Fig. 7 Sample images of medical dataset. Left: Original image. Right: Reconstructed image.

Subsequently, we proceeded to the second approach, which is based on Algorithm 1.The dataset, which originally consisted of only 2750 images, has been expanded to 5000 images by applying data augmentation techniques. Each class now contains 1000 augmented images. Then, we processed the dataset according to the data preprocessing steps outlined in Algorithm 1, and completed the cross-validation experiments. The cross-validation loss under multi-token discretization is shown in Figure 8(Mid).

Finally, we conducted experiments on the core theoretical aspects based on this dataset. The results, shown in Figure 8(Left), indicate that when the number of discrete tokens reaches 32, the overall discrete interactive communication outperforms continuous interactive communication, the former's average loss is 7.1% lower than that of the latter. The experimental results on the new dataset provided strong evidence to support our conclusions. Communication between agents who are unfamiliar with each other using multi-token discretized information variables is better than using continuous variables.

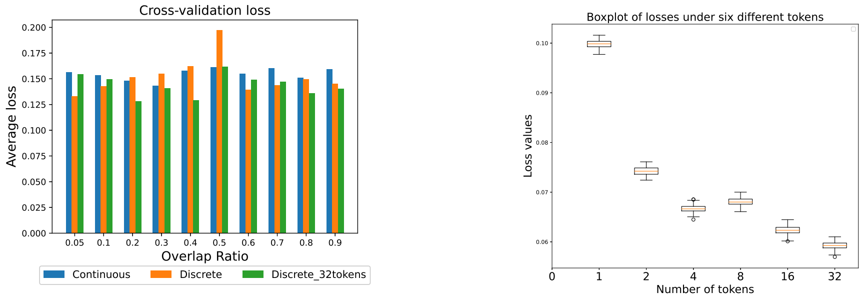

Fig. 8 Experimental results under the medical scenario dataset. Left: Communication loss on three types of models(AutoEncoder, Vector Quantized Variational Autoencoder, Vector Quantized Variational Autoencoder-32token); Mid: Communication losses of multiple agents under multi-token discretization. Right: Test loss during cross-training of multiple agents under multi-token discretization.

4.3 Research on codebook aspects

The experimental results above prove the core theory of this paper: when unfamiliar agents communicate with each other, the use of multi-token discrete semantics is more effective than that of continuous semantics. The subsequent research will mainly focus on the usage patterns of the codebook when communicating with discrete semantics and how to improve the codebook to enhance communication efficiency. In the research, the MNIST and CIFAR10 datasets are primarily used for exploration. Firstly, We investigated the impact of the size of the latent codebook space on the efficiency of discrete communication. We conducted experiments using a single-token VQVAE model following Algorithm 1, with the number of agents m set to 10. In the experiments, we controlled the experimental variable to be the size of the first dimension of the latent space. The result is shown in Figure 9. It can be observed that as the

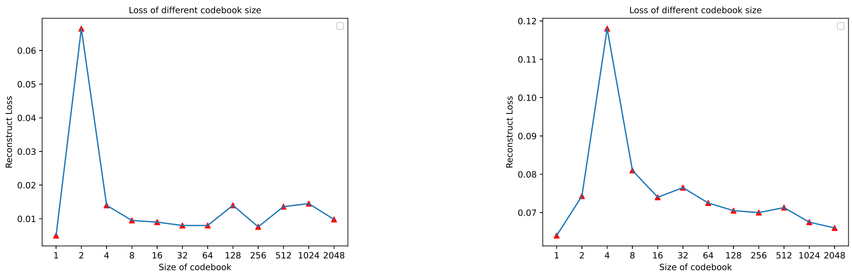

Fig. 9 The communication test losses under different latent space sizes. With the increase in the size of the codebook, the loss obtained by the 10 pairs of agents during the cross-validation phase shows a downward trend. Left: MNIST. Right: CIFAR10.

codebook space increases, the communication loss between different agents decreases overall. Although there is some fluctuation in the subsequent data for the MNIST dataset, we speculate that this is due to the small dataset size and the large codebook space, which results in the model not fully learning and the codebook not being evenly distributed. Therefore, we have reason to believe that as the codebook space expands, agents can capture more patterns when learning the language, thereby further improving the efficiency of discrete communication and enhancing the performance of discrete learning.

After completing the aforementioned experiments, in order to further investigate this direction in-depth, we conducted a study on the utilization of the codebook and some patterns in the VQVAE model. In the following results, our experiments were not conducted according to the aforementioned algorithm, but rather using a single model trained on the official datasets. Figure 11(left) represents the number of times

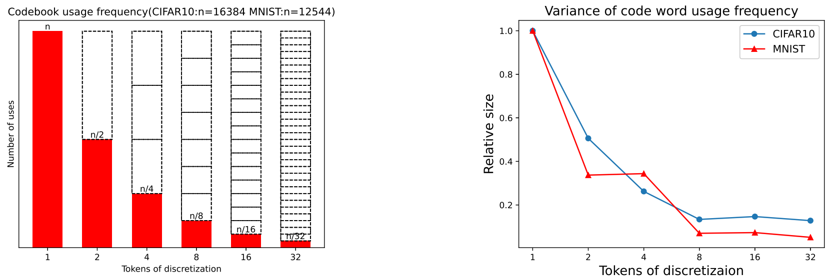

Fig. 10 The pattern of codebook usage with different numbers of discrete tokens. The number of times a codebook is used strictly follows the rules based on the number of tokens, and multi-token discretization facilitates the full utilization of the codebook.

code vectors are used for each codebook update. Assuming the single-token model uses a code vector N times for each codebook update. For an m-token model, each codebook update occurs $\frac{N}{m}$times. In each iteration, the codebook is updated m times, so after implementing multi-token discretization, the codebook updates strictly follow the rule based on the number of tokens, with the total usage of code vectors in each iteration remaining N, and any m-token model updating the codebook m times within that iteration, each token using $\frac{N}{m}$ code vectors. Figure 10(right) represents the variance between the frequencies of use of different code words for different numbers of discrete tokens. It can be observed that as the number of discrete tokens increases, the codebook is utilized more evenly. Figure 11 shows the proportion of the number

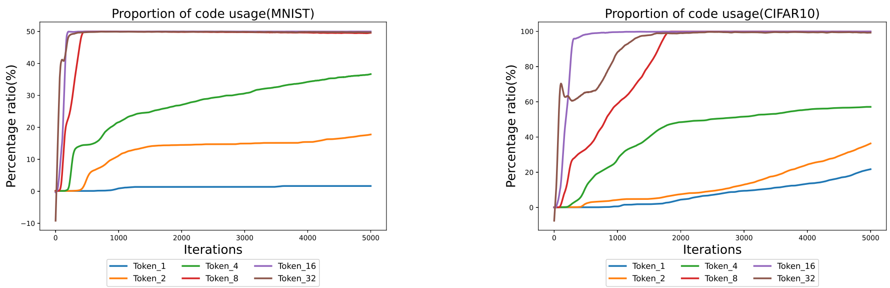

Fig. 11 Utilization rate of the codebook during the training iterations. Multi-token discretization facilitates the full utilization of the codebook.

of codewords used in each iteration to the total number, under different numbers of discrete tokens. It can be observed that when the number of discrete tokens is greater than or equal to 8, the codebook is effectively utilized throughout the iterations. This undoubtedly contributes to improving the model's performance. Figure 12 shows the transformation of codebook quantization loss for different numbers of discrete tokens. As the number of training iterations increases, the codebook's quantization loss gradually stabilizes, and a higher number of discrete tokens results in a higher stable loss

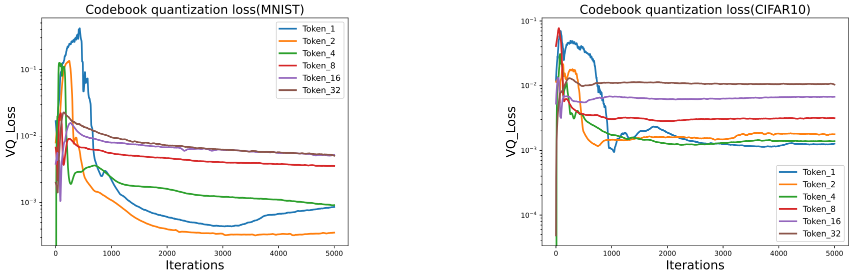

Fig. 12 Codebook quantization loss under different numbers of discrete tokens during training. Although multi-token discretization may cause an increase in quantization loss, the codebook is evidently learned more evenly.

value. For m agents with an overlap rate of 0.1, we explore the codebook similarity

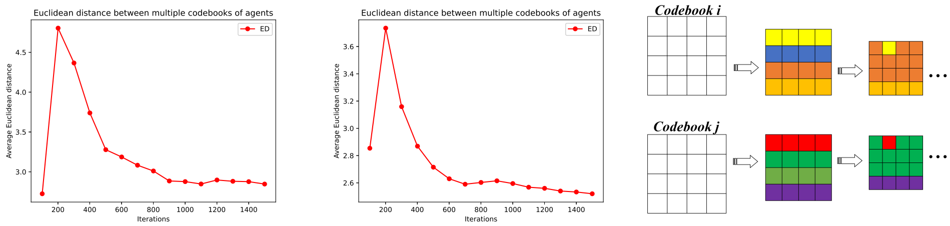

Fig. 13 The extent of difference in understanding of the same language among ten pairs of agents. Section 3 mentions that the understanding of the same language by different agents is not completely identical. However, as learning progresses, their similarity in language comprehension increases. Left: Experiments on MNIST. Mid: Experiments on CIFAR10. Right: The diagram to show that different codebook becomes more and more similar. That is, the codebook distributions in the figure are becoming increasingly similar.

among them during the training process, using the Euclidean distance as a measure. Assuming a codebook size of $\mathbb{R}^{L\times M}$, the calculation of the Euclidean distance is shown in Equation 6.

$$\mathbb{ED}_{Average} = \frac{1}{m(m-1)/2}\sum_{i=1}^{m}\sum_{j=i+1}^{m}\sqrt{\sum_{u=1}^{L}\sum_{v=1}^{M}(C_i(u,v)-C_j(u,v))^2} \tag{6}$$

C_i, C_j represents the codebook of different agents. Figure 13 illustrates the Euclidean distances between pairwise codebooks of ten agents during the learning process, with an overlap of 0.1. As the iterative learning progresses, the Euclidean distances between the latent codebooks of different agents decrease, indicating an increase in their similarity.

In section 4.3 of the research content, it is stated that increasing the size of the latent codebook space is beneficial for agents to improve their discrete communication efficiency. The patterns of codebook usage will aid in our future research endeavors. Especially the multi-token discretization mechanism, which has improved the issue of uneven codebook usage and mitigated the ”discretization bottleneck”.

5 Conclusion, Limitation and Future Study

Our experiments show that while the efficacy of communication between agents using single-token discrete semantics can rival that with continuous semantics, multi-token discretization before communication notably enhances the quality of information exchange over continous language. Furthermore, multi-token discretization outperforms single-token in terms of system generalization, revealing two key insights: multi-token communications are more effective than single-token approaches, and in contexts where agents encounter diverse languages, sentence construction with multiple discrete tokens yields optimal communication efficiency among isolated intelligent Agents. Additionally, we have put forth a theoretical foundation that contrasts the use of a VAE model against an AE model, noting that agent communicates through the VQVAE model falter in comparison to those facilitated by the VAE model, specific details will be given in the Appendix. The multi-token discretization method we proposed does not perform as well as the VAE, which may be due to optimization difficulties. Identifying the underlying reasons for this discrepancy will be the focus of our future research.

In summary, the multi-token discretization approach we propose outperforms the original single-token discretization method, and compared to continuous language based on the AE model, linguistic communication using multi-token discretization offers a greater advantage for communication among isolated agents.

References

[1] David H Ackley, Geoffrey E Hinton, and Terrence J Sejnowski. A learning algorithm for boltzmann machines. *Cognitive science*, 9(1):147–169, 1985.

[2] Anirudh Goyal ALIAS PARTH GOYAL, Aniket Didolkar, Nan Rosemary Ke, Charles Blundell, Philippe Beaudoin, Nicolas Heess, Michael C Mozer, and Yoshua Bengio. Neural production systems. *Advances in Neural Information Processing Systems*, 34:25673–25687, 2021.

[3] Diane Bouchacourt and Marco Baroni. How agents see things: On visual representations in an emergent language game. *arXiv preprint arXiv:1808.10696*, 2018.

[4] Rahma Chaabouni, Eugene Kharitonov, Emmanuel Dupoux, and Marco Baroni. Anti-efficient encoding in emergent communication. *Advances in Neural Information Processing Systems*, 32, 2019.

[5] Rahma Chaabouni, Eugene Kharitonov, Emmanuel Dupoux, and Marco Baroni. Communicating artificial neural networks develop efficient color-naming systems. *Proceedings of the National Academy of Sciences*, 118(12):e2016569118, 2021.

[6] Rahma Chaabouni, Florian Strub, Florent Altche, Eugene Tarassov, Corentin Tallec, Elnaz Davoodi, Kory Wallace Mathewson, Olivier Tieleman, Angeliki

Lazaridou, and Bilal Piot. Emergent communication at scale. In *International conference on learning representations*, 2021.

[7] Jakob Foerster, Ioannis Alexandros Assael, Nando De Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. *Advances in neural information processing systems*, 29, 2016.

[8] Washington Garcia, Hamilton Clouse, and Kevin Butler. Disentangling categorization in multi-agent emergent communication. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4523–4540, 2022.

[9] Allen Gersho and Robert M. Gray. Vector quantization and signal compression. In *The Kluwer International Series in Engineering and Computer Science*, 1991.

[10] Anirudh Goyal, Aniket Didolkar, Alex Lamb, Kartikeya Badola, Nan Rosemary Ke, Nasim Rahaman, Jonathan Binas, Charles Blundell, Michael Mozer, and Yoshua Bengio. Coordination among neural modules through a shared global workspace. *arXiv preprint arXiv:2103.01197*, 2021.

[11] Anirudh Goyal, Alex Lamb, Jordan Hoffmann, Shagun Sodhani, Sergey Levine, Yoshua Bengio, and Bernhard Scholkopf. Recurrent independent mechanisms. *arXiv preprint arXiv:1909.10893*, 2019.

[12] Shangmin Guo, Yi Ren, Serhii Havrylov, Stella Frank, Ivan Titov, and Kenny Smith. The emergence of compositional languages for numeric concepts through iterated learning in neural agents. *arXiv preprint arXiv:1910.05291*, 2019.

[13] Serhii Havrylov and Ivan Titov. Emergence of language with multi-agent games: Learning to communicate with sequences of symbols. *Advances in neural information processing systems*, 30, 2017.

[14] Sepp Hochreiter and Jurgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[15] Eugene Kharitonov, Rahma Chaabouni, Diane Bouchacourt, and Marco Baroni. Entropy minimization in emergent languages. In *International Conference on Machine Learning*, pages 5220–5230. PMLR, 2020.

[16] Simon Kirby. Natural language from artificial life. *Artificial life*, 8(2):185–215, 2002.

[17] Yair Lakretz, German Kruszewski, Theo Desbordes, Dieuwke Hupkes, Stanislas Dehaene, and Marco Baroni. The emergence of number and syntax units in lstm language models. *arXiv preprint arXiv:1903.07435*, 2019.

[18] Alex Lamb, Di He, Anirudh Goyal, Guolin Ke, Chien-Feng Liao, Mirco Ravanelli, and Yoshua Bengio. Transformers with competitive ensembles of independent mechanisms. *arXiv preprint arXiv:2103.00336*, 2021.

[19] Angeliki Lazaridou, Alexander Peysakhovich, and Marco Baroni. Multi-agent cooperation and the emergence of (natural) language. *arXiv preprint arXiv:1612.07182*, 2016.

[20] David Lewis. Convention harvard university press. *Cambridge, MA*, 1969.

[21] Sheng Li, Yutai Zhou, Ross Allen, and Mykel J Kochenderfer. Learning emergent discrete message communication for cooperative reinforcement learning. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 5511–5517. IEEE, 2022.

[22] Dianbo Liu, Alex M Lamb, Kenji Kawaguchi, Anirudh Goyal ALIAS PARTH GOYAL, Chen Sun, Michael C Mozer, and Yoshua Bengio. Discrete-valued neural communication. *Advances in Neural Information Processing Systems*, 34:2109–2121, 2021.

[23] Ryan Lowe, Abhinav Gupta, Jakob Foerster, Douwe Kiela, and Joelle Pineau. On the interaction between supervision and self-play in emergent communication. *arXiv preprint arXiv:2002.01093*, 2020.

[24] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

[25] Hua Miao and Nanxiang Yu. Targeted multi-agent communication with deep metric learning. *Engineering Letters*, 31(2), 2023.

[26] Yaru Niu, Rohan R Paleja, and Matthew C Gombolay. Multi-agent graph-attention communication and teaming. In *AAMAS*, pages 964–973, 2021.

[27] Emanuele Pesce and G. Montana. Improving coordination in small-scale multi-agent deep reinforcement learning through memory-driven communication. *Machine Learning*, 109:1727 – 1747, 2019.

[28] Cinjon Resnick, Abhinav Gupta, Jakob Foerster, Andrew M Dai, and Kyunghyun Cho. Capacity, bandwidth, and compositionality in emergent language learning. *arXiv preprint arXiv:1910.11424*, 2019.

[29] Jason Tyler Rolfe. Discrete variational autoencoders. *arXiv preprint arXiv:1609.02200*, 2016.

[30] David Simoes, Nuno Lau, and Reis. Multi-agent actor centralized-critic with communication. *Neurocomputing*, 390:40–56, 2020.

[31] Luc Steels. The synthetic modeling of language origins. *Evolution of communication*, 1(1):1–34, 1997.

[32] Olivier Tieleman, Angeliki Lazaridou, Shibl Mourad, Charles Blundell, and Doina Precup. Shaping representations through communication: community size effect in artificial learning systems. *arXiv preprint arXiv:1912.06208*, 2019.

[33] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[35] Kyle Wagner, James A Reggia, Juan Uriagereka, and Gerald S Wilkinson. Progress in the simulation of emergent communication and language. *Adaptive Behavior*, 11(1):37–69, 2003.

Appendix

Here we provide additional details about the experimental setup and additional results.

Algorithm 2 Cross-training Process

Require: Train m pairs of agents simultaneously;
Initialize encoders $E = \{e_0, ..., e_m\}$
Initialize codebooks $C = \{c_0, ..., c_m\}$
Initialize decoders $D = \{d_0, ..., d_m\}$

for each iteration i **do**
  Sample input data x_i
  Randomly sample encoder e_i, d_k, $c(i \neq k$, c is c_i or c_k)
  $h_i \Leftarrow e_i(x_i)$
  $z_i, L_{quantify} \Leftarrow c_i(h_i)$
  $x_i^{'} \Leftarrow d_i(z_i)$
  $L_i \Leftarrow L_i(x_i^{'}, x_i) + L_{quantify}$
  optimize e_i, c_i and d_i with respect to L_i
  for each pair of agents j **do**
    Sample validate data x
    $loss_j \Leftarrow d_j(c_j(e_j(x)))$
    output $loss_j$
  end for
end for

In Figure 6, we have demonstrated that the multi-token discretization mechanism is more effective in terms of communication between agents compared to the single-token discretization mechanism. Prior to this, we had already conducted some experimental work to prove the feasibility of the multi-token discretization mechanism. Algorithm 2 is the process through which we validate the effectiveness of multi-token discrete communication. We varied the intermediate processing architecture between the speaker and listener and recorded the test loss of m pairs of agents throughout the entire training process, as shown in Figure 1.

The curves in the figure indicate that when using cross-training, multi-token discretization indeed outperforms the single-token approach. Furthermore, the results demonstrate a pattern where increasing the number of tokens leads to better performance and faster learning. AE model still exhibits the fastest learning speed. That is, when paired agents learn a language, those that adopt a continuous semantic approach learn the fastest. However, as shown in Figure 5, when these agents interact with new agents, the outcomes are not as good as those of agents that learned through a discrete semantic approach.

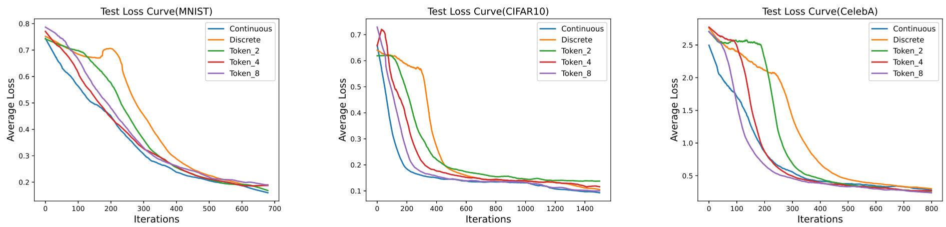

Fig. 1 Internal test loss during cross-training of multiple agents under multi-token discretization. Our results indicate that multi-token discretization has improved the learning efficiency of agents for communication languages.

In section 4, we mentioned the configuration of experimental parameters. For the four datasets, we adjusted the batch size or the size of the latent code space accordingly. However, all parameters for experiments within the same dataset must remain consistent. During our experiments, we attempted to use the VAE model instead of the AE model as a baseline, and simulated the learning and communication process between a pair of agents using continuous semantics. Similarly, we used Equation 5 to allocate individual datasets to each pair of agents, and the loss during training of the VAE model is represented by Equation 1.

$$\mathcal{L}_{VAE} = \mathcal{L}_{recon} + \beta\mathcal{L}_{KL} = \|x - x'\|_2 + \frac{\beta}{2}\sum_{j=0}^{J}\left(1 + log\left(\sigma_j^2\right) - \mu_j^2 - \sigma_j^2\right) \quad (1)$$

The first term represents the reconstruction loss, the second term represents the KL divergence loss, which characterizes the difference between the actual distribution of variables in the latent space and the prior distribution (usually assumed to be a standard normal distribution). Here, μ and σ respectively denote the mean and standard deviation of this distribution, while β represents a hyperparameter.

Regarding all the experiments on the autoencoder (AE) model, we replaced it with a variational autoencoder (VAE) model and repeated the experiments. When repeating the core experiments of Algorithm 1, we found that the agents learning with discrete variables did not achieve very good results when communicating with each other. That is, the loss from communication was relatively high. In contrast, the agents using continuous semantics for communication showed higher efficiency in their exchanges, with lower communication losses.

To further explore and compare the performance of continuous semantic communication based on the VAE model and discrete semantic communication, we have devised a series of experimental setups (see Figure 2). According to these setups, we conducted experiments, where the first major category of experiments involved the speaker's output being processed discretely for half of the information and the other half either being processed continuously or masked as zero. The second major category involved one half of the information being processed either discretely or continuously, while the other half was masked as zero. The experimental results indicate that regardless

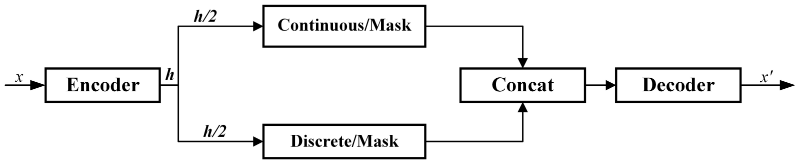

Fig. 2 Different methods used by paired agents during training and learning involve transmitting half of the information through continuous language, transferring the other half through discrete language, or masking one half of the information while delivering the remaining half using either continuous or discrete language. There are two models that simulate continuous language: AE and VAE.

of whether pairs of agents are trained and learned through the aforementioned combined model, or learned with singel model by masking half, under the condition of a lower overlap ratio, the relationship between the three types of cross-validation losses is: $AE > VQVAE > VAE$

However, this pattern is not absolute. Our main evaluation metric is the reconstruction loss between unfamiliar agents, as we mentioned in Section 3, where unfamiliarity indicates that their training datasets are not completely identical. When the proportion of overlap in the datasets is low, the discrete communication approach indeed does not perform as well as the continuous communication approach. Figure 3 represents some results when we trained using a combination of continuous and discrete methods, and used different validation methods during communication validation. The training method here involves splitting the encoder's output into two parts: one part goes through the latent variable layer of the VAE model, and the other part goes through the codebook layer of the VQVAE model, and then both parts are integrated into the decoder. During communication validation, we only use continuous, discrete, or a combination of continuous and discrete methods for cross-validation. The loss during the training process can be represented by Equation 2.

$$\mathcal{L} = \mathcal{L}_{recon} + \mathcal{L}_{quantization} + \mathcal{L}_{KL} \tag{2}$$

In this experiment, we found that when the overlap ratio is low, the effect of discrete communication is not as good as continuous communication. However, when the overlap ratio exceeds 90%, agents learning through discrete communication overcome the problems in learning communication protocols, leading to a reduction in overall communication loss and outperforming continuous methods. However, the results of this experiment were obtained considering that both continuous and discrete information

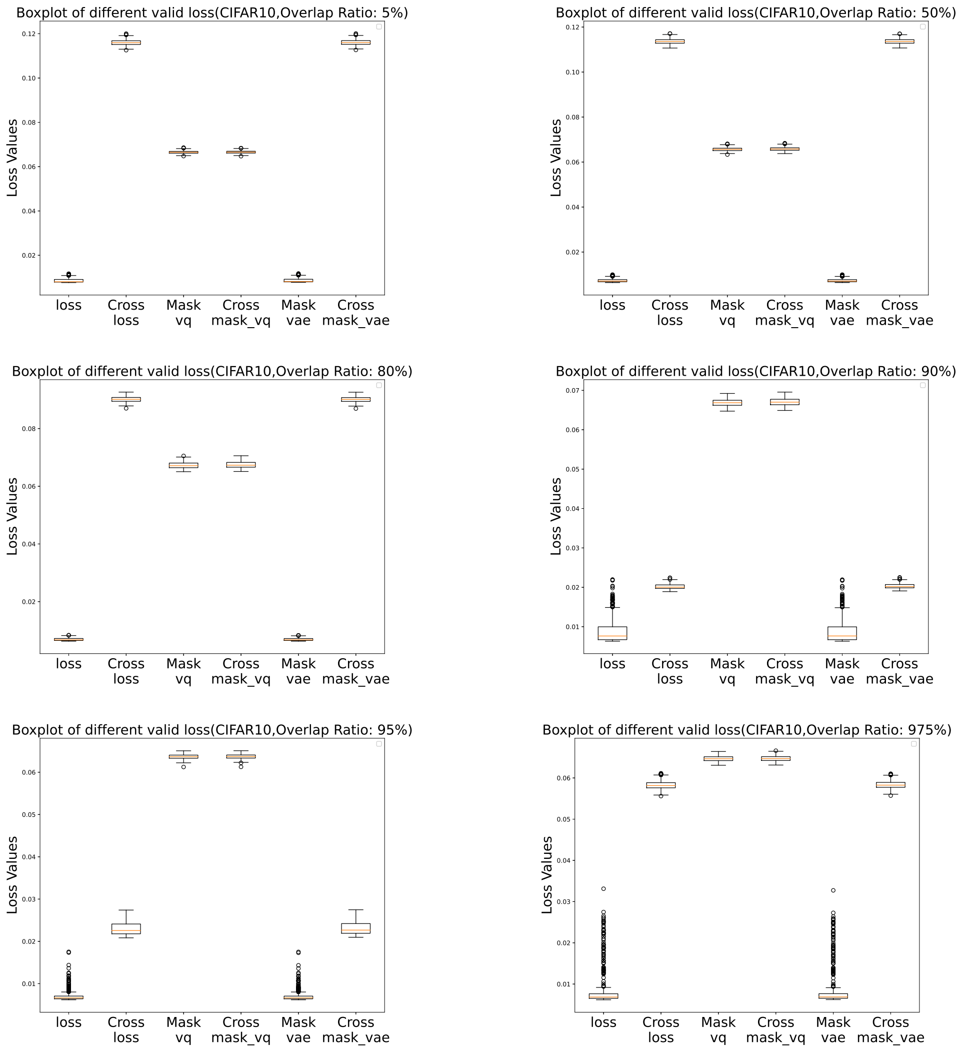

Fig. 3 The loss incurred by agents communicating in different ways. During the training process, a combination of continuous and discrete methods is used. During the validation phase, we employ three methods to obtain communication loss: the first way is to mask the content of the continuous information part and only use the discrete information part for communication; the second is to mask the content of the discrete information part and only use the continuous information part for communication; the third is to use both parts of information for integrated communication. When the familiarity between different pairs of agents exceeds 90%, the effectiveness of communication using discrete semantics surpasses that of using continuous semantics.

are present in the communication process of agents. When the agents learn and communicate entirely in a discrete or continuous manner, the advantage of the discrete method also disappears even with a 90% overlap ratio.

Regarding the experiments on non-image datasets and the lack of demonstrated advantages over variational autoencoders, we will reserve them for continued research.