Discrete Messages Improve Communication Efficiency among Isolated Intelligent Agents

Anonymous authors
Paper under double-blind review

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.

1 Introduction

Intelligent agent communication, positioned at the intersection of AI and linguistics, explores the development of a shared language among agents. The Lewis Game(Lewis, 1969) serves as a key example of collaborative tasks in this area, where a speaker and listener work together to identify a specific object from a set of alternatives (Figure 1(left)). This field has seen extensive research into the origins and evolution of language through AI, with studies investigating various aspects of emergent communication. Chaabouni et al. (2019) focused on the compositionality and generalization capabilities of language agents and the creation of efficient color naming systems(Chaabouni et al., 2021a). Lazaridou et al. (2016) proposed blending multi-agent communication with data-driven natural language learning to facilitate machine-human interaction. Lowe et al. (2020) explored the interplay between supervised learning and self-play in developing communication protocols for emergent communication, while deep reinforcement learning has been applied to tackle the challenges of multi-agent communication(Simoes et al., 2020; Foerster et al., 2016), especially as the number of agents increases, leading to potential redundancy and inefficiency in communication. Recent research(Niu et al., 2021; Resnick et al., 2019; Chaabouni et al., 2021b; Bouchacourt & Baroni, 2018) has focused on optimizing communication strategies, expanding our understanding of language evolution by addressing both capacity and dataset complexity challenges. Vector Quantization (VQ), in line with Shannon's rate-distortion theory(Gersho & Gray, 1991), suggests that vector encoding can outperform scalar encoding by effectively handling dependencies in source symbols. Recent advancements in reparameterization, particularly for VAEs managing discrete variables(Rolfe, 2016; Maddison et al., 2016), have enhanced model effectiveness. The VQ-VAE(Van Den Oord et al., 2017) model overcomes non-differentiability issues by employing the identity function for efficient gradient transmission, proving to be more robust and generalizable than continuous counterparts for complex learning models(Van Den Oord et al., 2017; Liu et al., 2021). Moreover, the application of discretization in multi-agent reinforcement learning tackles communication challenges within modular reasoning architectures, facilitating efficient interactions across modules(Goyal et al., 2019).

This study builds on previous research by advancing the vector quantization technique in VQ models. Our hypotheses suggest that when intelligent agents have diverse personal experiences, communicating via dis-

crete messages, especially through sentences composed of multiple discrete tokens, is more effective than using continuous messages. Our experiments in multi-agent machine learning have empirically demonstrated that communication through multi-token discrete sentences significantly enhances communication efficiency among agents with diverse experiences. However, when employing the VAE model to simulate continuous language communication between agents, rather than the AE model, its effectiveness surpasses that of multi-token discrete language communication.

The structure of our paper is organized as follows: Section 1 provides a background on agents' communication and the objectives of our research. Section 2 introduces related work in the field. Section 3 details the experimental settings and methods employed in our study. In Section 4, we present our experimental results and draw meaningful conclusions. Finally, Section 5 concludes our research with discussions and prospects for future work.

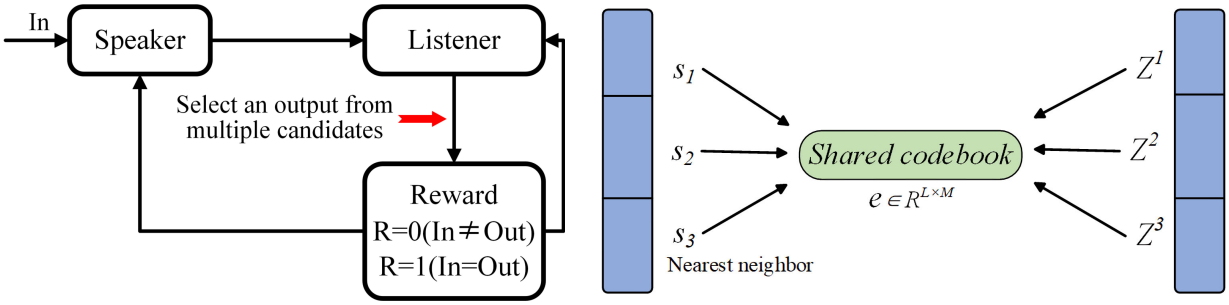

Figure 1: Left: Lewis Game. Right: Multi-token Discrete Mechanism. The communication vector is initially divided into multiple discretization tokens. Each token goes through separate discretization, where it is quantized to the nearest neighbor within a shared collection of latent codebook vectors. Subsequently,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

2 Related Work

In recent years, several methods have been devised to improve communication within machine learning models, notably through attention mechanisms(Goyal et al., 2019; ALIAS PARTH GOYAL et al., 2021; Goyal et al., 2021) and the Transform method(Vaswani et al., 2017; Lamb et al., 2021). Furthermore, collective memory and shared parameters have enhanced multi-agent communication(Pesce & Montana, 2019). The Reinforced Inter-Agent Learning (RIAL) model stands as a prominent framework for discrete communication among intelligent agents, supporting the use of discrete symbols to enable interactions reminiscent of human social behaviors(Foerster et al., 2016). The work of Guo et al. (2019) delves into the crucial role of computer simulations in evolutionary linguistics, illustrating how intelligent agent models can foster the development of compositional languages for numerical concepts through communication. Research into LSTM(Hochreiter & Schmidhuber, 1997) language models by Lakretz et al. (2019) has illuminated the manner in which hidden states encapsulate numerical values and syntactic structures, spurring further exploration into linguistic patterns. Subsequent studies(Miao & Yu, 2023; Garcia et al., 2022; Havrylov & Titov, 2017) have expanded our comprehension of the dynamics of multi-agent communication and the genesis of language among neural network-based agents.

Building on this foundation, our research leverages the Vector-Quantized Variational Autoencoder (VQ-VAE) model(Van Den Oord et al., 2017) and adopts cross-training and cross-validation techniques to scrutinize communication patterns between agents. Our findings reveal that in environments where agents employ diverse language systems, discrete forms of language are more efficacious than continuous ones. We also delve into how the variation in token numbers within codebooks affects the efficiency of discrete communication, a subject that receives in-depth treatment in Section 4.

3 Theoretical Basis and Experimental Method

In the series of autoencoder models(Ackley et al., 1985), we designate the encoder component of the model as the speaker, while the decoder component is referred to as the listener. The communication between the speaker and the listener within the same model is considered internal communication, while the communication between the speaker or listener and other model components is referred to as agents' communication. We define $e(\cdot)$ as the encoder function, $d(\cdot)$ as the decoder function, and $h(\cdot)$ as the quantization layer function of VQ-VAE. The codebook space of the quantization layer is denoted as $C = [c_1, c_2, ...c_n]$.

3.1 Discrete and Continuous Communication

For the discrete communication model, we use VQ-VAE. 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 $Z = h(z_e(x), \varphi)$. 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 quantization loss in the VQ-VAE model, in the subsequent algorithm, we use $L_{quantify}$ to represent these two items. 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. And the involvement of the codebook quantization layer mentioned above refers to the process of discrete 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 communication is less effective than discrete communication between unfamiliar agents.

3.2 Multi-token Discretization

Li et al. (2022) 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. The VQ-VAE model builds upon the AE and introduces a latent space codebook between the encoder and decoder. 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: $e_{o_i} = h(s_i)$, where $o_i = \underset{j}{argmin} \left\| s_i - c_j \right\|_2$. After the discretization process, the N segments of data are then integrated back together in the order of their original splitting: $Z = CONCAT(e_{o_1}, e_{o_2}, ...e_{o_N})$. 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). Since we divide the data into N segments, the total loss function for model training is defined as shown in Equation 3.

$$\mathbb{L} = \mathbb{L}_{task} + \frac{1}{N}\left\{\sum_{i=1}^{N} ||sg[s_i] - e_{o_i}||_2^2 + \beta \sum_{i=1}^{N} ||s_i - sg[e_{o_i}]||_2^2\right\} \tag{3}$$

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

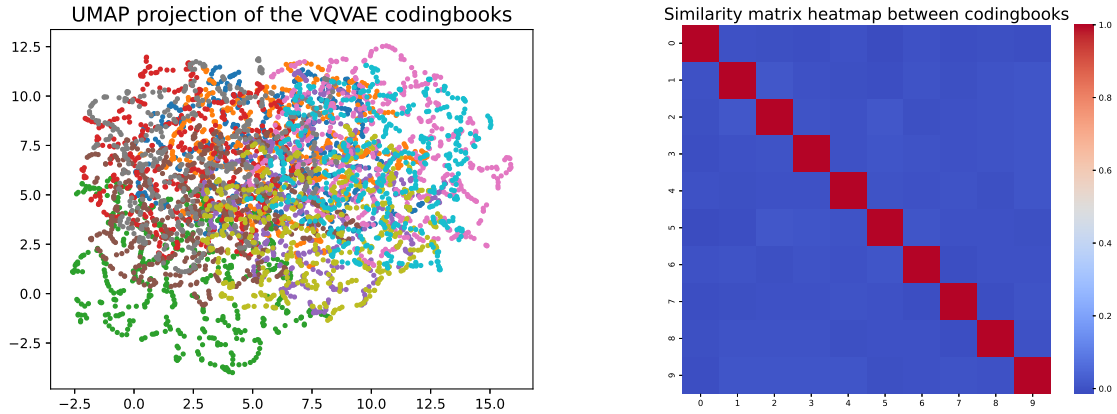

Figure 2: Ten agents' understanding of the same language. Left: Feature distribution of latent codebooks for different agents. Right: Similarity of different latent codebooks. The understanding of this language is different for each agent.

3.3 Learning and Validation of Communication for Agents

Attempts have been made to explore cross-training(Guo et al., 2019; Tieleman et al., 2019) 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 agents trained simultaneously on the same MNIST dataset. Each color represents a internal communication protocol, that is, the feature distribution in the codebook. It can be observed that there are differences in semantic understanding among 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 A implements the aforementioned process.

Algorithm 1 Individual Training and communication

Using the processed dataset: $train_set = (train_set_1, train_set_2, ...train_set_m)$
Train j agents simultaneously
Initialize encoders $E = \{e_0(\cdot), ..., e_m(\cdot)\}$.
Initialize quantization layers $H = \{h_0(\cdot), ..., h_m(\cdot)\}$
Initialize decoders $D = \{d_0(\cdot), ..., d_m(\cdot)\}$
for each iteration i **do**
    for each agent j **do**
        Sample input data x_j from $train_set_j$
        $L_j \Leftarrow MSE(x_j, d_j(h_j(e_j(x_j)))$
        optimize $e_j(\cdot)$,$h_j(\cdot)$ and $d_j(\cdot)$ with respect to L_j
    end for
end for
Validation:
for each agent j **do**
    Sample validate data x from val_set
    $loss_j \Leftarrow MSE(x, d_k(h_j(e_j(x)))), (k = 1, 2...m, k \neq j)$
    output $loss_j$
end for
$Communication\ loss = \sum loss_j$

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 $val_set = (sample_1^1, sample_2^1, ...sample_m^1)$. 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 $Overlapset = (sample_1^2, sample_2^2, ...sample_m^2)$. 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 4.

$$\frac{m * sample_i^2}{P_{train} + (m-1) * sample_i^2} = p_j, (i = 1, 2..., m) \tag{4}$$

$Overlapset$ were then merged with the respective training sets $dataset_i^{'}$, where $dataset_i^{'} = dataset_i - sample_i^1 - sample_i^2$, ensuring that each class in the training set contained images from the remaining $m-1$ classes. The processed training set is $train_set = (train_set_1, train_set_2, ...train_set_m)$. This process resulted in a training set where each category served as a separate training set for single agent to learn from. The experiments conducted on the split dataset follow Algorithm 1, which forms the core of our research paper. 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 VQ-VAE model. However, when we incorporate the AE model in our experiments, we simply remove quantization layers H from the procedure, and the data outputted by the encoder is directly decoded by the decoder.

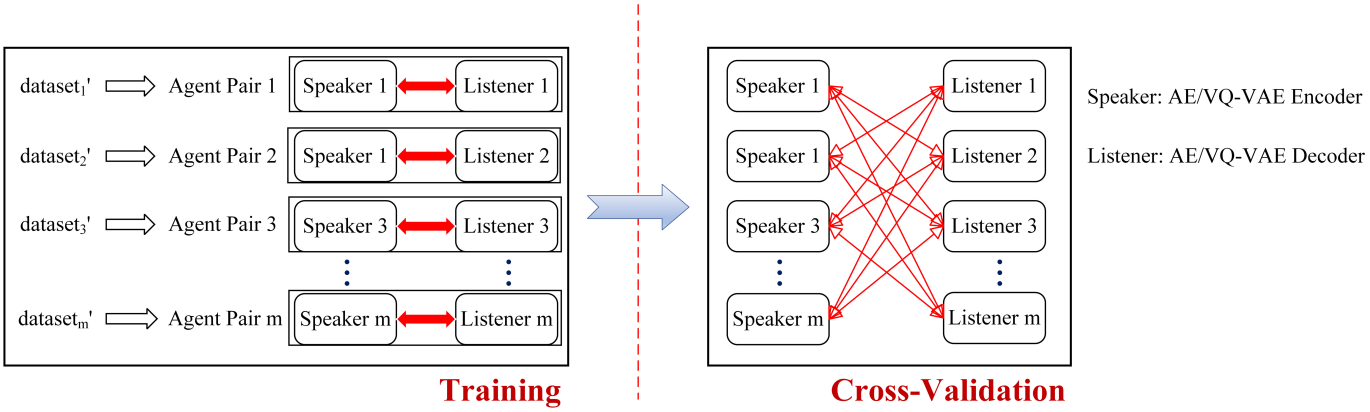

Figure 3: Training and validation of agents. Each agent has its own dataset during training. Upon completion of learning, one agent interacts with the other agent . This is our core methodology, where in this validation scenario, the advantage of discrete language in communication between agents is determined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

4 Experiments

In our work, we employed four datasets: MNIST, CIFAR10, CelebA and Diabetic Retinopathy dataset. The image resolution for all four 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 evaluate the effectiveness of agents' communication by measuring the error between the original images and the reconstructed images. An example of the two types of images can be seen in Figure 4.

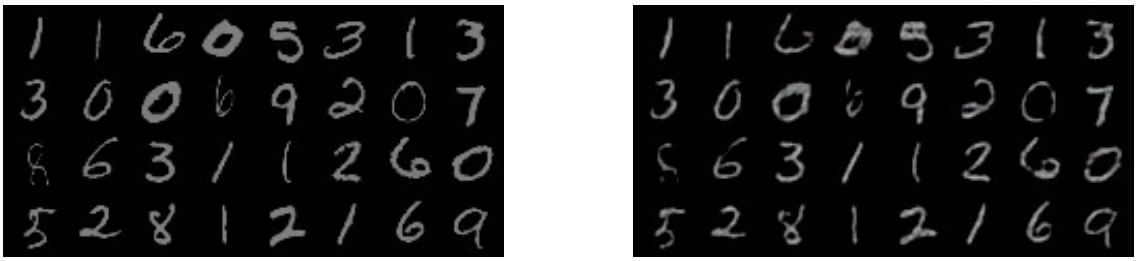

Figure 4: Example images for the reconstruction task. Left: Original image. Right: Reconstructed image.

4.1 Multi-token Discretization for Improved Agents' Communication

According to the method shown in Figure 3 and Algorithm 1, we repeated the experiments with different overlap ratios using the multi-head discretized VQ-VAE model with the best performance and the AE model. For the MNIST and CelebA datasets, the original VQ-VAE model had latent space size $C_m \in \mathbb{R}^{512\times 64}$ and $C_m \in \mathbb{R}^{512\times 128}$, while for the CIFAR10 dataset, it was $C_m \in \mathbb{R}^{1024\times 256}$. 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 6

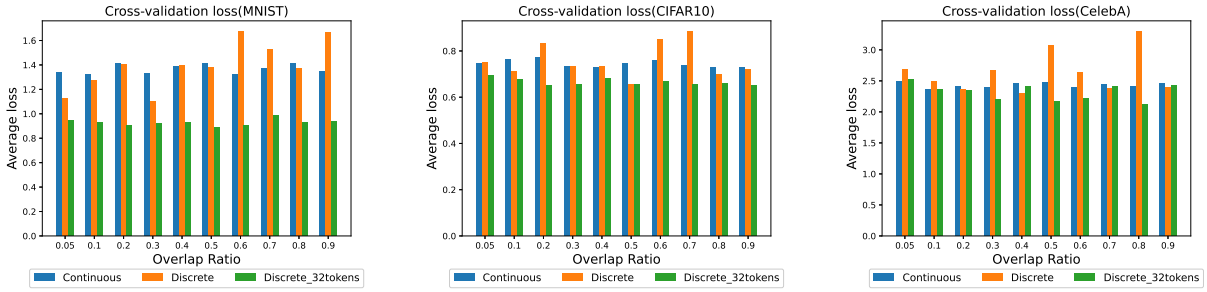

Figure 5: Communication loss on three types of models. The x-axis represents different overlap ratios, and the y-axis represents the communication loss between agents.

explain why we chose to conduct our experiments with a 32-token VQ-VAE model 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 agents simultaneously according to Algorithm 1, where each boxplot in the figure represents the stable loss from communications between the m agents. The general pattern is that as the number of discrete tokens increases, the communication loss decreases.

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

4.2 Theoretical Validation and Practical Application

Based on the open-source datasets, we conducted the same experiments as in section 4.1 with Diabetic Retinopathy dataset(Figure 7) to further validate our theory. When experimenting with the VQ-VAE model on this dataset, codebook size $e_m \in \mathbb{R}^{512\times 128}$. The dataset is divided into 5 categories based on symptom

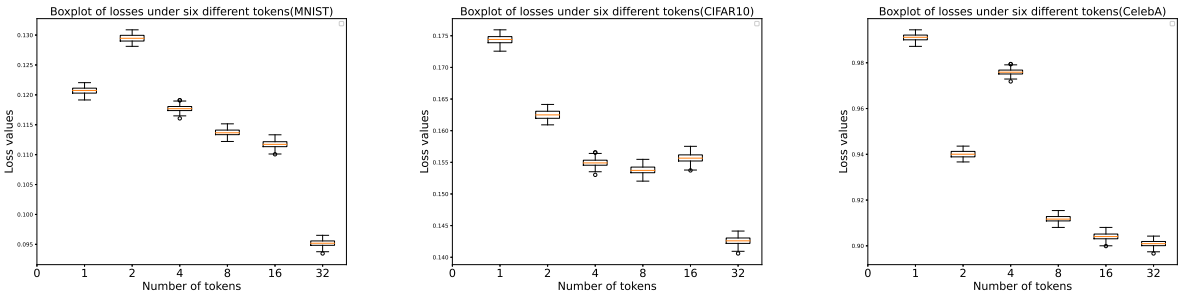

Figure 6: The communication loss of multiple agents with the same data overlap ratio under multi-token discretization. The increase in the number of tokens can reduce the loss from communications.

types, with varying numbers of images in each category. We performed 5 sets of experiments, each involving communications between agents.

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

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. First, we processed the dataset according to the data preprocessing steps outlined in Algorithm 1, and completed the communication-validation experiments. The communication loss under multi-token discretization is shown in Figure 8(Right).

Then, 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.

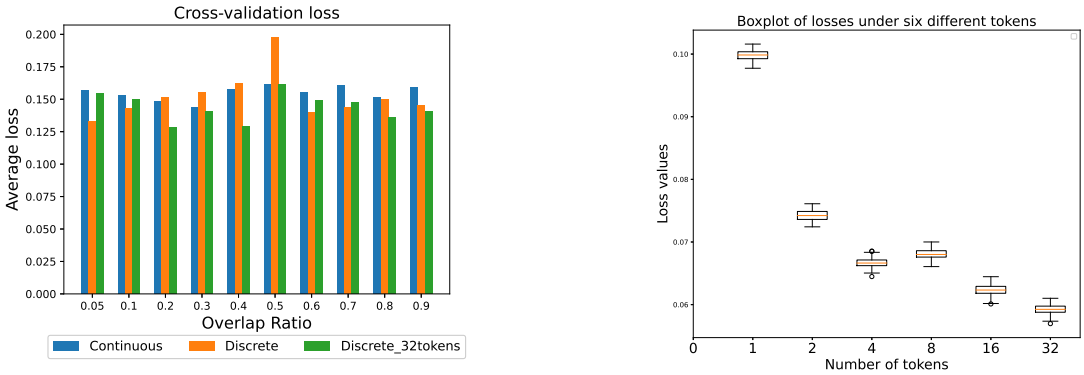

Figure 8: Experimental results under Diabetic Retinopathy dataset. Left: Communication losses on three types of models(AE, VQ-VAE, VQ-VAE-32token); Right: Communication losses of multiple agents under multi-token discretization.

4.3 Research on codebook aspects

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 VQ-VAE 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. Although there is some fluctuation in the subsequent data for the MNIST

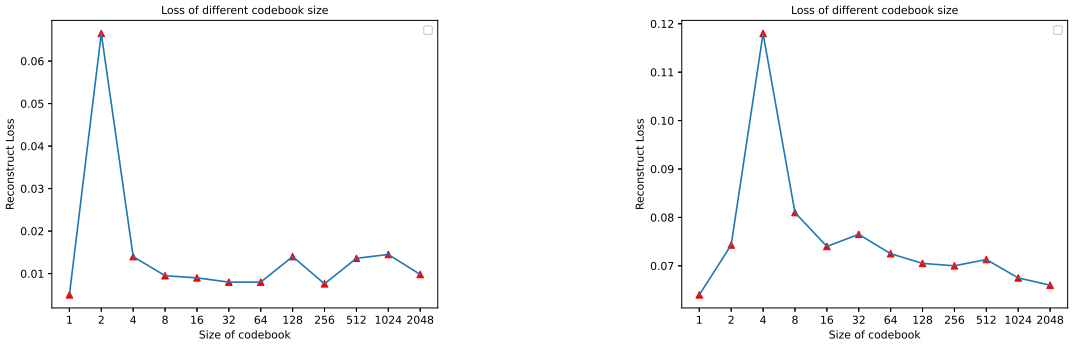

Figure 9: The communication losses under different latent space sizes. With the increase in the size of the codebook, the loss obtained by 10 agents during the communication-validation phase shows a downward trend. Left: MNIST. Right: CIFAR10.

dataset, we speculate that this is due to the small dataset size and the large codebook space. 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.

In order to further investigate this direction in-depth, we conducted a study on the utilization of the codebook and some patterns in the VQ-VAE 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 10(Left) represents the number of times code vectors are used for each codebook update. Assuming

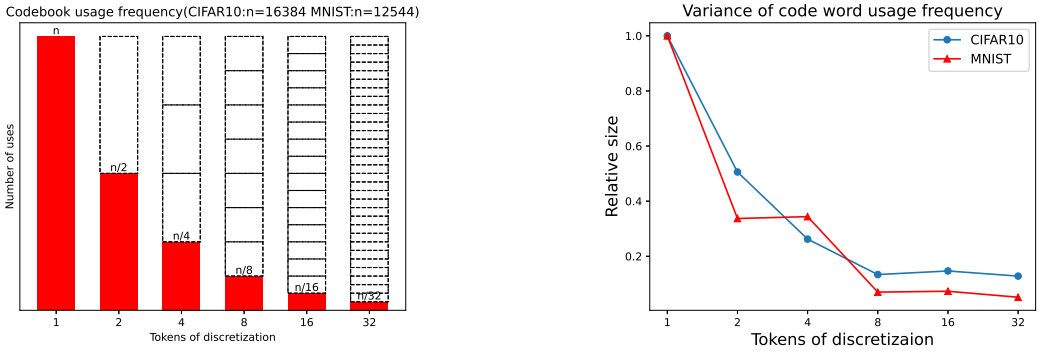

Figure 10: The pattern of codebook usage with different numbers of discrete tokens. The number of times a codebook is used strictly follows the rules based on the number of tokens, and multi-token discretization facilitates the full utilization of the codebook.

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 of codewords used in each iteration to the total number,

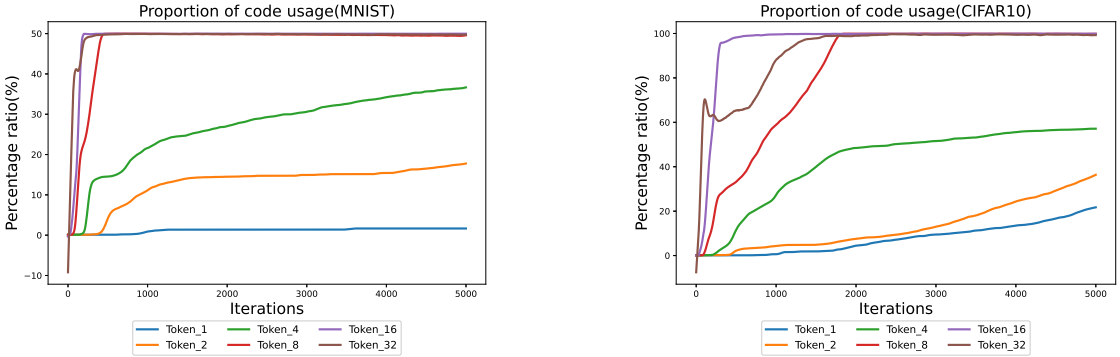

Figure 11: Utilization rate of the codebook during the training iterations. Multi-token discretization facilitates the full utilization of the codebook.

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. Figure 12 shows

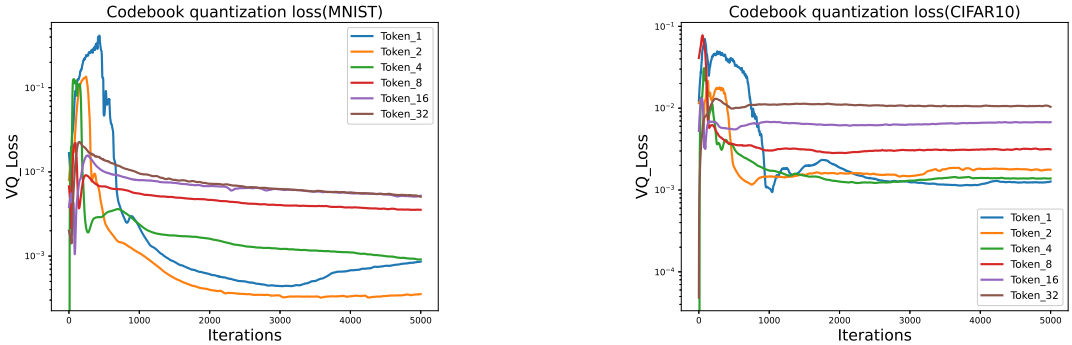

Figure 12: Codebook quantization loss under different numbers of discrete tokens during training. Although multi-token discretization may cause an increase in quantization loss, the codebook is evidently learned more evenly.

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 value. For m agents with an overlap rate of 0.1, we explore the codebook similarity 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 5.

$$\mathbb{ED}_{Average} = \frac{1}{m(m-1)/2}\sum_{i=1}^{m}\sum_{j=i+1}^{m}\sqrt{\sum_{u=1}^{L}\sum_{v=1}^{M}(C_i(u,v)-C_j(u,v))^2} \tag{5}$$

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

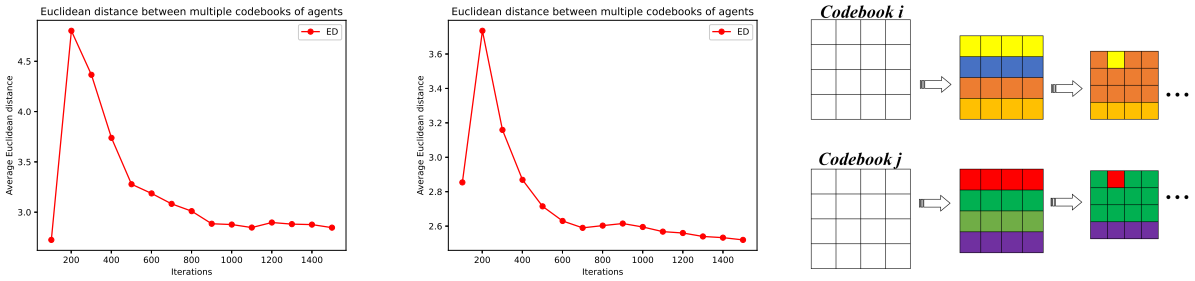

Figure 13: The extent of difference in understanding of the same language among 10 agents. As learning progresses, their similarity in language comprehension increases. Left: Experiments on MNIST. Mid: Experiments on CIFAR10. Right: The diagram to show that different codebook becomes more and more similar.

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 have shown that communication between agents using single-token discrete semantics can achieve comparable efficacy to that of continuous semantics. However, multi-token discretization before communication significantly enhances the quality of information exchange compared to continuous language. Additionally, multi-token discretization outperforms single-token approaches in terms of system generalization. This suggests that multi-token communications are more effective, especially in contexts where agents encounter diverse languages. Furthermore, we have explored the use of a VAE model versus an AE model for communication, and found that agents' communication facilitated by the VAE model outperforms VQ-VAE model-based communication. We aim to investigate the underlying reasons for this discrepancy in future research. In summary, the multi-token discretization approach we propose outperforms the original single-token discretization method, and compared to continuous communication based on the AE model, using multi-token discretization offers a greater advantage for communication among isolated agents.

Author Contributions

Acknowledgments

References

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

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.

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

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

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, 2021a.

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*, 2021b.

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.

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*, pp. 4523–4540, 2022.

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

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

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.

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.

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.

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

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.

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.

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

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

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)*, pp. 5511–5517. IEEE, 2022.

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.

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.

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.

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

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

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.

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.

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

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

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.

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

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.

A Appendix

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

Algorithm 2 Cross-training Process

Train m agents simultaneously;
Initialize encoders $E = \{e_0, ..., e_m\}$
Initialize quantization layers $H = \{h_0(\cdot), ..., h_m(\cdot)\}$
Initialize decoders $D = \{d_0, ..., d_m\}$
for each iteration i **do**
    Sample input data x
    Randomly sample encoder e_i, d_k, $c(i \neq k$, c is c_i or c_k)
    $z_q \Leftarrow e_i(x)$
    $z_i, L_{quantify} \Leftarrow h_i(z_q)$
    $x' \Leftarrow d_i(z_i)$
    $L_i \Leftarrow MSE(x', x) + L_{quantify}$
    optimize e_i, h_i and d_i with respect to L_i
    for each pair of agents j **do**
        Sample validate data v
        $loss_j \Leftarrow d_j(h_j(e_j(v)))$
        output $loss_j$
    end for
end for

The CelebA dataset we used in all experiments was 8000 images extracted from official sources, and in the experiment of Algorithm 1, we divided them into 8 categories based on their attributes for use by 8 agents. 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. For the dataset used in Algorithm 2, MNIST and CIFAR10 are used directly without any changes, while the CelebA and retinal datasets are divided into two categories: training and validation sets. That is, the agents in the algorithm are exposed to the same training or validation sets. We varied the intermediate processing architecture between the speaker and listener and recorded the test loss of m agents throughout the entire training process, as shown in Figure 14.

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. That's why we initially wanted to use multi-token discretization mechanism for agents' communication. AE model still exhibits the fastest learning speed. That is, when agent 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.

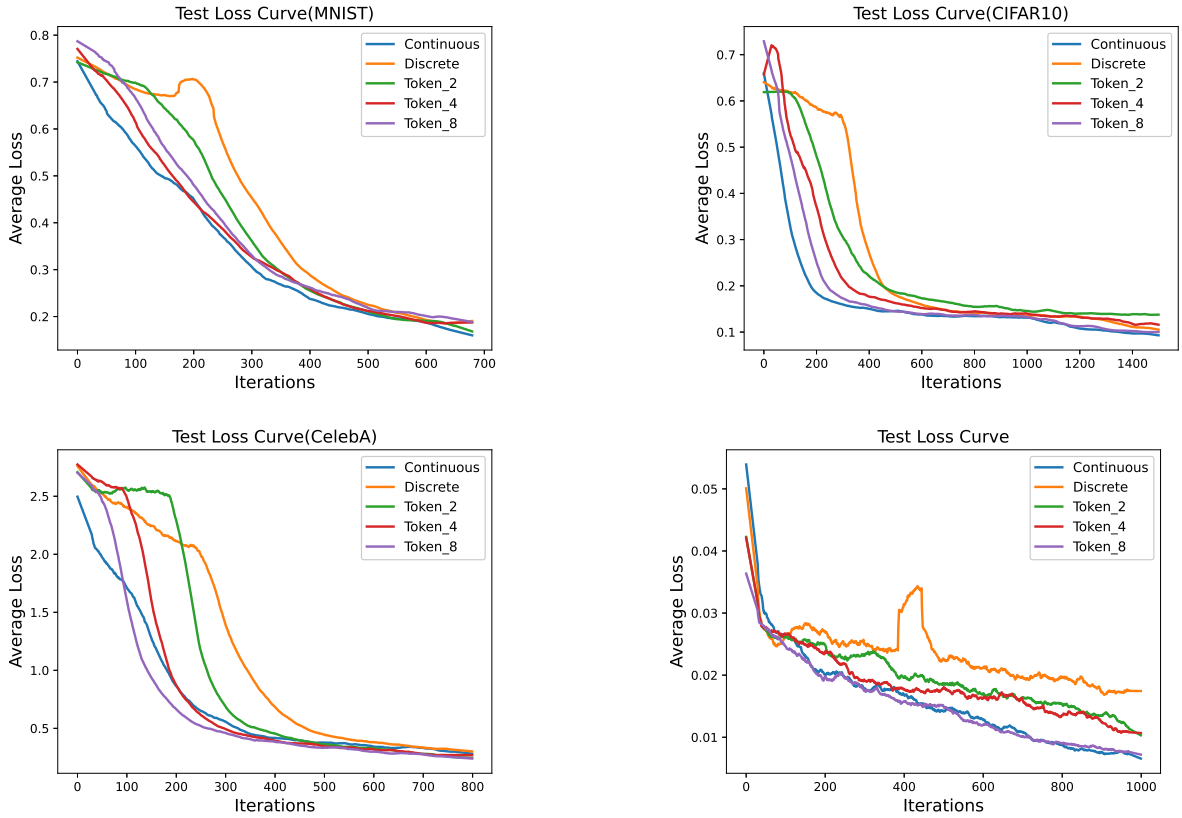

Figure 14: Internal test loss during cross-training of multiple agents. Compared to the single token, multi-token discretization has improved the learning speed of agents for communication languages.

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 4 to allocate individual datasets to each agent, and the loss during training of the VAE model is represented by Equation 6.

$$\mathcal{L}_{VAE} = \mathcal{L}_{recon} + \beta\mathcal{L}_{KL} = \|x - x'\|_2 + \frac{\beta}{2}\sum_{j=0}^{J}\left(1 + log\left(\sigma_j^2\right) - \mu_j^2 - \sigma_j^2\right) \tag{6}$$

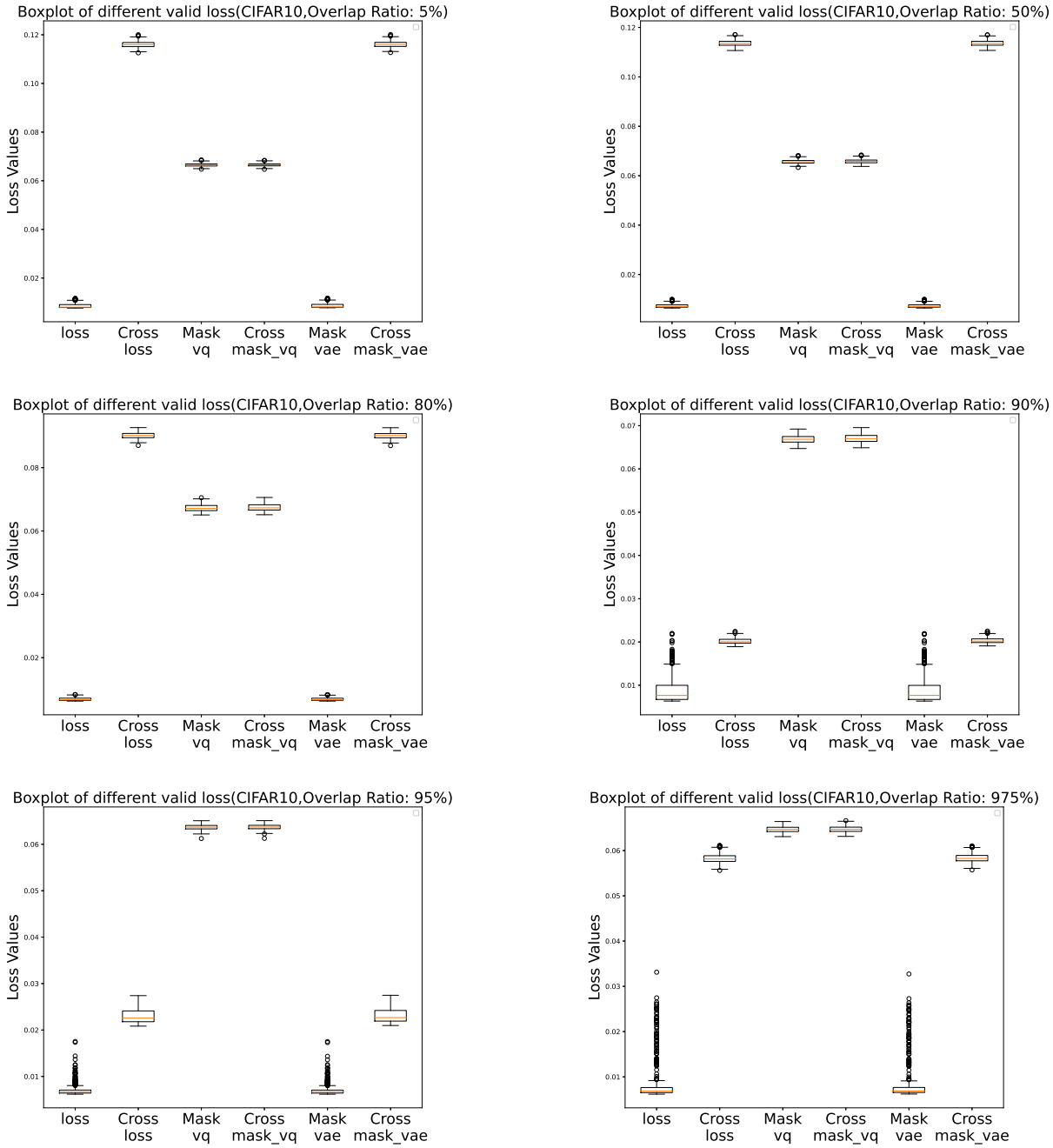

Figure 15: The loss incurred by agents communicating in different ways. During the training process, a combination of continuous and discrete methods is used. During the validation phase, we employ three methods to obtain communication loss: the first way is to mask the content of the continuous information 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.

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 loss.

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 16). 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. The second major category involved one half of the information being processed either discretely or continuously, while the other half was masked as zero. We have conducted extensive experiments on the structure of Figure 16, but still have not yielded good results. The experimental results indicate that regardless of whether 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 performance of discrete communication methods is indeed inferior to continuous communication methods based on VAE. Figure 15

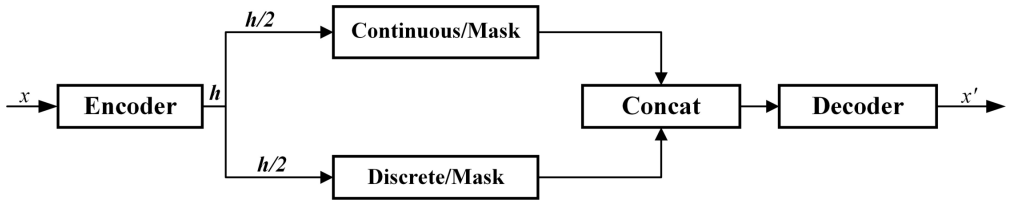

Figure 16: Different methods for communication between agents. One method is to perform continuous communication on half of the information and discrete communication on the other half. Another method is to mask half of the information and use the remaining half for continuous or discrete communication. Here, two models are used to simulate continuous communication, namely autoencoder (AE) and variational autoencoder (VAE).

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 VQ-VAE 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 7.

$$\mathcal{L} = \mathcal{L}_{recon} + \mathcal{L}_{quantization} + \mathcal{L}_{KL} \tag{7}$$

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 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.