ISSN 2706-655X Vol.6, Issue 5: 67-78, DOI: 10.25236/IJFET.2024.060510

Robust Reinforcement Learning for Robotic Manipulation Using Lite ViT and Saliency Maps

Hao Guo[1,2], Xiulai Wang[1,*], Ningling Ma[1], Yutao Zhang[1]

[1]Nanjing Jinling Hospital, Affiliated Hospital of Medical School, Nanjing University, Nanjing, China
[2]School of Computer Science and Technology, Harbin Institute of Technology, Harbin, 150000, China
[*]Corresponding author

Abstract: *The advancement of reinforcement learning has revolutionized autonomous robotic manipulation, enabling robots to handle complex tasks efficiently. This study presents an innovative RL framework integrating a Transformer-based visual encoder, leveraging lite Vision Transformers and saliency maps to enhance general performance across multiple robotic manipulation tasks in the OpenAI Gym environment. Our approach employs a three-channel encoder to extract and integrate visual information from multiple perspectives, bolstering visual robustness and task performance. Additionally, we incorporate unsupervised learning, data augmentation, and Soft Actor-Critic methods to ensure data efficiency during training. Experimental results demonstrate that our method achieves superior success rates, particularly in the Push task, underscoring the efficacy of saliency maps in robotic vision tasks. This research is pivotal for applications requiring robust multi-task performance with limited training data.*

Keywords: *robotics; reinforcement learning; vision transformer; saliency maps*

1. Introduction

The evolution of reinforcement learning (RL) has transformed how robots autonomously handle complex manipulative tasks. This technological advancement shines in its deployment across various sectors, including digital gaming [1-4], intricate behavioral mimicry like miming leg motions [5,6], and the mastery of interactions with objects through grasping [7,8] and adjusting their orientation [9]. Such innovations have been proven effective in both synthetic setups and tangible environments. Additionally, RL that integrates visual inputs capitalizes on intricate data, thereby crafting more dynamic handling solutions that are likely viable in practical settings. These systems discern and react to the surroundings via direct sensory feedback, consciously avoiding traditional methods of state information retrieval [10-14].

In the realm of RL with a focus on visual inputs, current methodologies primarily enhance data utilization but often downplay the crucial factor of policy durability. Notable methodologies like Sim2Real [7,15,16], imitation learning [17-20], and data augmentation [21,22] have been implemented to tackle data insufficiency in both theoretical and practical robotic contexts, yielding early advantageous outcomes. Despite these advancements, the refinement of policy efficacy is still required. Commonly, robotic manipulation tasks exhibit inconsistent performances, posing substantial challenges in evening out these discrepancies. The performance of RL strategies, which depend heavily on visual data, is critically dependent on the integrity of these inputs, shaping both the developmental trajectory and final achievements. Stability in policy, influenced by variability in task execution under different scenarios, is crucially dependent on visual inputs. Additionally, task distinction within varied visual scenes remains a key determinant of success. This paper emphasizes the need for an advanced image encoder in maintaining a robust feedback-based system, adept at isolating essential features while minimizing irrelevant visual clutter, thus bolstering policy development and ensuring system integrity.

Contemporary RL frameworks frequently incorporate convolutional neural networks (CNNs) for extracting visual features. This fusion of ConvNets with RL strategies has proven efficient in enhancing performance on specific tasks [21–24]. Yet, CNNs require substantial data for training, which can negate the data efficiency originally intended by the system and increase the costs associated with refining the encoder.

In this paper, we propose an effective RL framework with a Transformer-based visual encoder that

ISSN 2706-655X Vol.6, Issue 5: 67-78, DOI: 10.25236/IJFET.2024.060510

significantly improves policy general performance for vision-based robotic manipulation with two camera views, taking advantage of the superiority of the lite ViT [25] and preserving the RL method efficiency. At the same time, saliency maps of eye-to-hand view are also attached as an additional input view. Specifically, each view is encoded individually using the single channel of the visual encoder in framework, features extracted from the ConvNet in encoder comprise spatial information naturally, casting off the extra positional embeddings in original ViT. Features from each channel are fused before entering the linear layer. Features fusion compensates for the spatial information demand of single view and unifies the visual attention in different views. After encoder procession, the visual features are utilized as the input of framework to optimize the policy end-to-end until convergence.

This paper presents an advanced RL framework employing a Transformer-based visual encoder. This setup markedly boosts the overall efficacy of policies for robotic manipulation using two camera perspectives, harnessing the streamlined capabilities of the lite ViT [25] while retaining RL methodological efficiency. Moreover, we integrate saliency maps from the eye-to-hand perspective as supplementary visual data. Distinctively, each viewpoint is processed separately through an individual channel of our visual encoder, where the ConvNet inherently captures spatial details, obviating the need for the usual positional embeddings of the traditional ViT. We then merge the feature sets from each channel prior to their introduction to the linear layer, thus enhancing spatial data integration and synchronizing visual focus across various angles. Subsequently, these enriched visual inputs guide the framework's policy optimization process through to full convergence.

Our methodology promotes data efficiency by fusing contrastive unsupervised learning with data augmentation techniques in an offline RL setting [24]. The initial step involves compiling ten demonstrations into a replay buffer. These demonstrations then serve to initialize the encoders through a method of contrastive unsupervised pre-training. In the final stage, the framework proceeds to train with this processed data, supplemented by additional images acquired during the active training phases.

Recent investigations have shown that combining egocentric and third-person camera perspectives, which provide complementary information from both close-up and wide views, is more effective than using a single fixed-lens monocular camera [7,24]. This finding supports Land et al.'s principle that the oculomotor system focuses on the target's location during complex tasks. Moreover, we have added saliency maps to boost and hasten attention, helping the framework quickly identify the essential features of the robot and the objects it interacts with, thereby enhancing learning efficiency.

To validate our approach, we conducted comprehensive testing in the OpenAI Gym Fetch environment, comparing our method with top ConvNet-based techniques such as FERM [24] and SVEA [8] across various tasks, including Reach, Push, and Pick-and-Place. Our approach successfully maintained data efficiency and improved overall performance. Remarkably, in the Reach task, our method achieved a success rate of 94% with the least number of training steps, surpassing the performance of FERM, SVEA (CNN encoder), and SVEA (ViT encoder), which had success rates of 65%, 78%, and 81%, respectively. The primary contributions of our work are: (i) The introduction of an efficient RL-based framework that enhances performance across three critical tasks; (ii) The development of a triple-streaming process architecture within the visual encoder that enables effective data fusion and utilizes triple-view information to improve manipulation precision.

2. Related literature

2.1 Data-Efficient Transformer

Since the introduction of attention mechanisms from language processing to computer vision, ViT and their variants have shown extraordinary results in vision tasks. However, their "data-hungry" nature still impedes data-efficient applications. Research on lightweight Transformers is burgeoning [26-29]. Touvron et al [30]. proposed the Data-Efficient Image Transformer (DeiT), which uses a novel knowledge distillation procedure to reduce ViT's data dependence, performing well on mid-sized datasets like ImageNet and fine-tuning on smaller datasets such as CIFAR-10 [31] for image classification. Furthermore, Sanh et al. [32] developed DistilBERT, which leverages knowledge distillation to create a smaller, faster model while maintaining performance. Lan et al. [33] introduced ALBERT, which employs parameter sharing to reduce the number of parameters significantly. These advancements demonstrate that Transformers can achieve state-of-the-art accuracy on small datasets, comparable to leading CNNs, while mitigating data dependence. Our work also leverages the method architecture of [25] as the basic visual module and illuminates the general performance and data efficiency of our

ISSN 2706-655X Vol.6, Issue 5: 67-78, DOI: 10.25236/IJFET.2024.060510

proposed encoder based on the Transformer.

2.2 Visual Learning for RL

Extensive research has been conducted to improve the data efficiency of vision-based RL [34-36]. Recent developments in this area have highlighted the importance of data augmentation and unsupervised representation learning. For example, recent models such as the EfficientNet [37] and EfficientDet [38] have been integrated with vision-based RL frameworks to enhance data efficiency. Similarly, Dosovitskiy et al. [39] developed ViT models that achieve competitive performance with fewer data requirements through improved architectural designs and training techniques. Despite these advancements, traditional CNN visual encoders still face challenges in consistently extracting object features due to data efficiency constraints. The integration of Transformers with RL [40-44], such as in the DeiT and ALBERT, has shown potential improvements in specific visual tasks. However, the overall performance of ViT in RL can be inconsistent, sometimes even underperforming compared to original visual encoders, mainly due to the extensive data requirements for general tasks. In this work, we address the "data-hungry" nature of ViTs while maintaining their high performance. Our approach includes the use of a specially designed visual encoder, unsupervised contrastive learning, and data augmentation techniques. Furthermore, we compare our method with state-of-the-art techniques and the original ViT in Section 4.2, demonstrating its potential for broader application in various robotic manipulation tasks.

2.3 Contrastive Learning

Contrastive learning, a self-supervised method, learns general representations by enforcing similarity constraints on datasets with pairs of similar and dissimilar points, using unlabeled data. In contrast to generative learning, which focuses on detailed pixel information, contrastive learning targets abstract information by minimizing distances between similar data clusters and maximizing distances between dissimilar ones. This approach has seen significant advancements in the field of computer vision in recent years [45, 46]. He et al. [39] introduced Momentum Contrast (MoCo), which maintains a large, consistent dictionary for contrastive learning, facilitating the use of larger batch sizes and stabilizing training. Additionally, Grill et al. [47] proposed BYOL (Bootstrap Your Own Latent), a technique that removes the need for negative samples and shows strong performance in learning representations. In our work, we employ instance-based contrastive learning and pre-training methods, which have shown effectiveness in both computer vision and reinforcement learning in simulated environments. This approach aims to enhance data efficiency and improve overall model performance by leveraging these cutting-edge techniques.

2.4 Data Augmentation

Data augmentation [48-50] is a technique that enhances model performance by modestly increasing the dataset size, thus improving general performance even with limited data. Visual augmentation, which is closely related to our work, has shown significant benefits. Laskin et al. [22] demonstrated that using data augmentations alone can greatly enhance the data efficiency and generalization of RL methods that rely on pixel inputs, without modifying the RL algorithms themselves. Similarly, Hansen et al. [8] found that applying data augmentation in deep Q-learning algorithms greatly improves both model stability and generalization. Research has consistently shown that random cropping significantly improves sample efficiency. However, while many data augmentation methods boost sample efficiency, they also pose a higher risk of divergence. In our approach, we incorporate random crop visual augmentation into our robotic manipulation framework to improve both data efficiency and model generation.

3. Method

To enhance data efficiency in agent training, we employ a framework comprising three integral modules: a demonstration buffer, contrastive unsupervised learning with a dual-streaming encoder, and reinforcement learning augmented with data enhancement techniques to achieve an efficient RL policy. An overview of the architecture is presented in Figure 1, with detailed descriptions provided below.

ISSN 2706-655X Vol.6, Issue 5: 67-78, DOI: 10.25236/IJFET.2024.060510

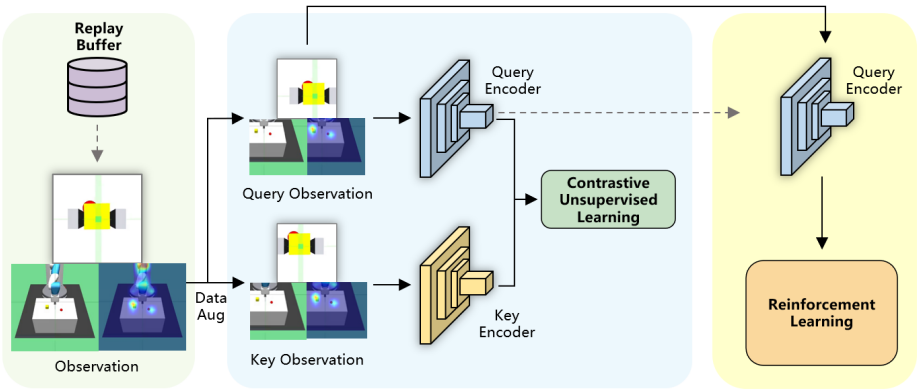


Figure 1: Architectural overview

3.1 Demonstration

Drawing from the research by Zhan et al. [24], the use of ten demonstrations has been identified as the optimal setting to balance efficiency and training performance. Our study adopts a similar approach, utilizing demonstrations for unsupervised contrastive pre-training. Consequently, we have configured the buffer to include ten demonstrations for each of the simulated Gym environments, namely Fetch Pick-and-Place, Fetch Reach, and Fetch Push.

3.2 Contrastive Unsupervised Learning

Once the replay buffer is initialized, we train the dual-streaming encoder using contrastive learning, a subset of representation learning. This method can be conceptualized as a differentiable dictionary look-up task. The query–key pairs in contrastive learning are generated via random crop data augmentation. These pairs, referred to as anchor and positive, aim to maximize agreement, whereas pairs involving positive and negative images aim to minimize agreement. Specifically, two different augmentations of the same image serve as the anchor and positive observations, with other images designated as negatives. In this system, given a query

q and a set of keys $K=\{k_0,k_1,\ldots\}$ K is divided into k^* and K/k^*. This setup ensures that q aligns better with k^* than with any other key k_i in K/k^*. Here, q, k^*, K, and K/k^* are synonymous with anchor, positive, target, and negative, respectively. The matching performance between q and k can be evaluated using dot products $q^T k$, bilinear products $q^T W k$, or Euclidean distances. In the unsupervised learning context, where datasets lack labels or prior knowledge to directly identify negatives, contrastive learning employs a loss function to maximize the score of the (q, k_+) pair while minimizing the score of the (q, k_-) pair. This method emphasizes critical features in similarity relations relative to negative pairs. InfoNCE, proposed by [51], combines Noise Contrastive Estimation (NCE) and mutual information optimization. It functions as a multi-class cross-entropy loss for a K-class softmax classifier, meeting the requirements of unsupervised contrastive learning. The infoNCE loss is defined as follows:

$$\mathcal{L}_q = \log \frac{exp(q^T W k_+)}{exp(q^T W k_+) + \sum_{i=0}^{K-1} exp(q^T W k_i)} \tag{1}$$

Results from FERM show that pretraining an image encoder made up of CNN layers does not significantly enhance task performance in both simulated environments and basic real-world scenarios. However, with the advent of Transformers as state-of-the-art models in NLP and CV, the integration of CV with Transformers suggests that ViTs may outperform traditional CNN models. Without the inductive biases of CNNs, such as translation equivariance and locality, it is thought that Transformer performance depends greatly on model size and the volume of training data. Considering the data-intensive nature of ViTs, which complicates data-efficient robot training, we developed a model inspired by [25]. This model is optimized for small datasets while maintaining state-of-the-art performance in image encoding, thus ensuring both efficiency and effectiveness during the pre-training phase.

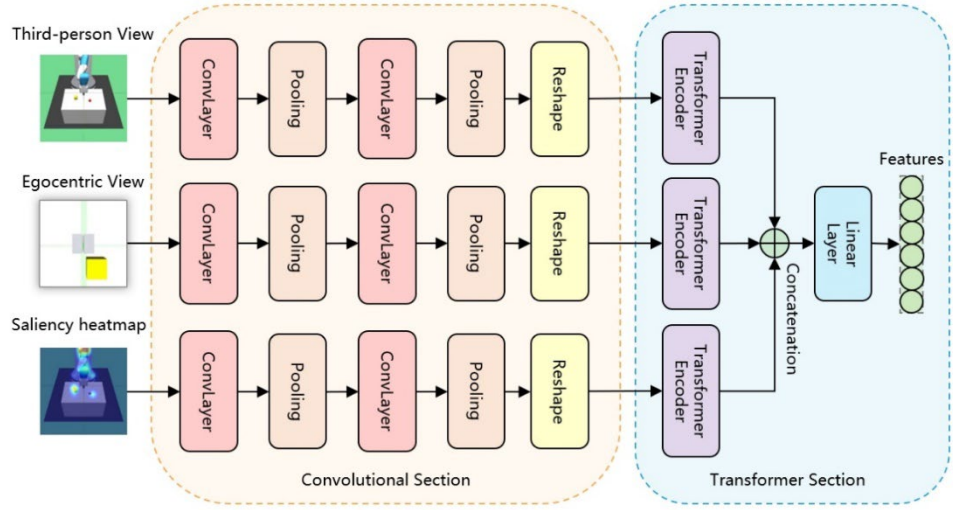
To optimize the lite ViT for our purposes, originally designed for classification in small datasets, we removed the SeqPool component, which maps sequential outputs to a single class index in the original model. Additionally, unlike previous methods that primarily fuse data from each channel early in the three-view process [7,24], we perform vector concatenation later in the visual encoder. This approach

ISSN 2706-655X Vol.6, Issue 5: 67-78, DOI: 10.25236/IJFET.2024.060510

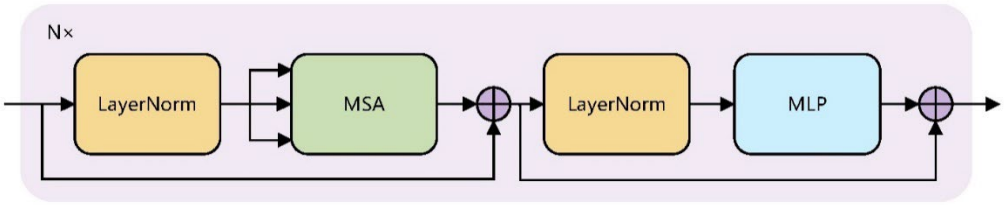
allows for more comprehensive extraction of individual features from each channel, retaining complementary information from all three views. By avoiding narrow dimensionality reduction or data filtration for a single channel, this method preserves the confidence contribution from each view and prevents data redundancy.

3.2.1 Encoder Architecture

Our lite ViT model integrates two main components: a convolutional section and a Transformer section. In the original ViT, images are split into patches and transformed into sequences for the Transformer, with positional embeddings providing necessary spatial information. According to [25], smaller patches improve performance in small datasets. To ensure efficient data training, we incorporated a convolutional section into the image encoder, integrating the attention mechanism within the CNN encoder. This method provides both local and spatial information about patch relationships without needing additional positional embeddings. As depicted in Figure 2, the image encoder processes three views through distinct channels. Images are divided into patches, tokenized through pooling and reshaping for the Transformer. The output vectors from each Transformer encoder are then concatenated into a single vector, with features extracted by the final linear layer. Figure 2a shows the visual encoder in our study, and Figure 2b depicts the Transformer encoder.



(a)



(b)

Figure 2: ViT encoder architecture

3.2.2 Convolutional Section

To retain the inductive bias properties of the original image encoder within the Transformer model, the convolutional section has been designed to handle image patching, positional embedding, and image embedding functions. The convolutional section employs a traditional architecture, utilizing a sequence of two convolutional layers with ReLU activation followed by a max pooling layer, repeated twice within each channel. the input image $i \in R^{C\times H\times W}$, the output o:

$$i_0 = MaxPool\left(ReLU\left(Conv2d(i)\right)\right) \tag{2}$$

To match the embedding dimension of the Transformer backbone, the number of the filters in conv2d is the same as the embedding dimension. At the end of the convolutional section, the max pooled data are reshaped in a sequence, which is proper to enter the Transformer section. The overlapping of the convolutional kernels stride increases the information density which is easily neglected when it appears at the boundary of the patches in original ViT, since the significant image information may be broken

ISSN 2706-655X Vol.6, Issue 5: 67-78, DOI: 10.25236/IJFET.2024.060510

and lost its original value after it is divided into two or more parts, even the addition of the positional embeddings cannot reveal the initial affection. In addition to this, overlapping can raise the performance to rely on the inductive bias. Compared to ViT, the utilization of the convolutional section divides the image into patches in a more flexible way, and the image is embedded into a potential representation that is more suitable and efficient for the Transformer encoder.

3.2.3 Transformer Section

The Transformer section features a Transformer encoder paired with a linear layer, based on the original ViT design. The encoder comprises a Multi-headed Self-Attention (MSA) layer and a Multi-Layer Perceptron (MLP) head. It utilizes layer normalization for data standardization, employs the GELU activation function, and includes residual connections to prevent degradation of the network. After the Transformer encoder processes the data, outputs from each channel are combined into a single tensor. This tensor then undergoes dimensionality reduction through a linear layer, generating features tailored for the RL algorithm.

3.3 Reinforcement Learning with Data Augmentation

After pretraining, the visual encoder provides processed data to the RL phase. While on-policy algorithms are popular due to their stability and potential, they require millions or even billions of actions, which is impractical for real-world robot training. In contrast, off-policy algorithms are better suited for training robots with pixel-based data in real environments, as they allow multiple reuses of samples without overfitting by not depending on the assumption that samples come from the current trained policy.

Soft Actor Critic (SAC), an off-policy algorithm, has proven effective in enabling a quadruped robot to learn to walk within 2 hours [6] and mastering manipulation skills in simulated gym environments [21–23]. Based on these successes, we utilize SAC as the agent to interact with the environment, incorporating augmentation techniques. Considering the efficiency and practicality of various augmentation methods in both simulated and real-world settings, we use random patches from the original frame as augmentation images for contrastive pre-training. During gradient updates, the agent leverages a combination of demonstration data and training data.

3.3.1 Soft Actor Critic

Soft Actor Critic is an advanced off-policy algorithm designed to optimize stochastic policies, maximizing expected rewards in continuous control environments. By utilizing data augmentation, SAC extends its efficiency from state-based observations to pixel-based observations [21,22,24], achieving performance comparable to state observations in simulated benchmarks. As an actor-critic method, SAC trains both a policy π_ϕ and two critics, $Q_{\phi 1}$ and $Q_{\phi 2}$. The critic parameters ϕ_i are optimized by minimizing the squared Bellman error:

$$\mathcal{L}(\phi_i, \mathcal{B}) = E_{t\sim\mathcal{B}}\left[\left(Q_{\phi_i}(o,a) - (r + \gamma(1-d)\mathcal{T})\right)^2\right] \tag{3}$$

where $t = (o, a, o', r, d)$ is the tuple with observation o, action a, reward r, and done signal d, is the replay buffer where the demonstration observations are stored, and T is the target that the critics are trained to match, defined as:

$$\mathcal{T} = \min_{i=1,2} Q^*_{\phi_i}(o', a') - \alpha \log \pi_\psi(a'|o') \tag{4}$$

where $Q^*_{\phi_i}$ is the exponential moving average (EMA) of the parameter of Q_{ϕ_i}. Empirical studies indicate that using the EMA can boost training stability in off-policy RL algorithms. The entropy coefficient α plays a role in determining the balance between entropy maximization and the primary optimization goal. To learn the desired actor policy, the actor stochastically samples from the policy π_ψ and is trained to maximize the expected return as follows:

$$\mathcal{L}(\psi) = E_{a\sim\pi}\left[Q^\pi(o,a) - \alpha \log \pi_\psi(a|o)\right] \tag{5}$$

where $a_\psi(o,\xi) \sim \tanh\left(\mu_\psi(o) + \sigma_\psi(o) \odot \xi\right), \xi \sim \mathcal{N}(0, I)$ is a standard normalized noise vector.

4. Experiments

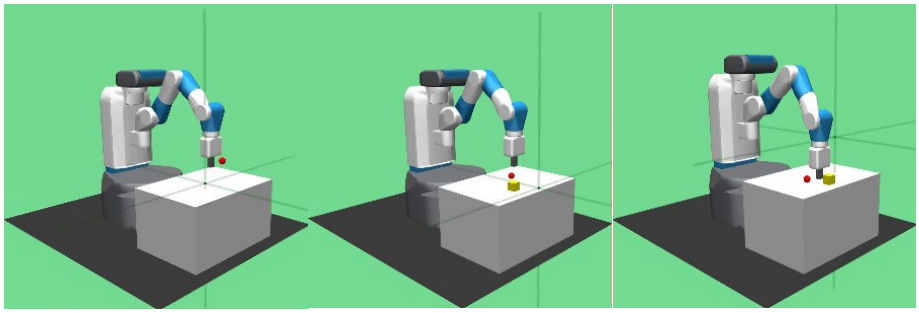
In this section, we explore the advantages of incorporating the attention mechanism within a pixel-based robotic manipulation training framework. This process includes handling image observations in

ISSN 2706-655X Vol.6, Issue 5: 67-78, DOI: 10.25236/IJFET.2024.060510

the visual encoder before they are utilized by the agent in the RL environment. We assess the performance and efficiency of our approach by comparing it to the baselines of FERM, SVEA (CNN encoder), and SVEA (ViT encoder). Furthermore, we perform ablation experiments to determine the impact of each component within our method.

4.1 Experiments Setup

We evaluate our proposed framework in the OpenAI Gym robotics environment, focusing on simulated goal-based tasks for Fetch robots. These tasks include Pick-and-Place, Push, and Reach. Compared to the task setup in FERM, the corresponding tasks are similar in our simulated environment. Although FERM's Pickup and Move tasks demonstrate good performance in the final evaluation, they require significantly longer optimization times, with the Move task taking the longest and having the lowest success rate. Figure 3 illustrates the three tasks in the OpenAI Gym Fetch environment. Additionally, we employ a combination of master and egocentric views as dual-camera perspectives, which is further discussed in Section 4.3.4.



(a) *(b)* *(c)*

Figure 3: Experimental environments. (a) Reach; (b) Pick-and-Place; (c) Push.

As noted in FERM, unsupervised pre-training did not yield significant performance benefits for simpler tasks in their suite. To explore and enhance the effectiveness of unsupervised pre-training, we applied it across various tasks. Additionally, we referenced the study by Hansen et al. [8], which introduced a general RL framework that stabilizes Q-value estimation under data augmentation using two data streams. SVEA also examined the use of CNN and ViT as visual encoders, achieving high data efficiency and performance, and reaching state-of-the-art (SOTA) results.

4.2 Results

In this section, we compare the results to highlight the performance and data efficiency from our investigation. We evaluate data efficiency using the number of interaction steps, instead of interaction time, to minimize the impact of CPU or GPU computational power. We assess overall training performance by the final success rate. Interaction steps include those needed for policy optimization, indicating the steps required to reach convergence, and those needed to achieve a 90% success rate, representing a high-performance level. Table 1 presents the comparison results. First, our proposed method shows significant improvements in performance and data efficiency across three simulated robotic manipulation environments compared to FERM, SVEA (CNN encoder), and SVEA (ViT encoder). Particularly for the Push task, our method achieves the highest final success rate and is the only method to reach a 90% success rate. For other tasks, our method not only matches the full success rate of other methods but also requires the fewest steps to converge, minimizing policy optimization time. Second, when comparing the steps needed to reach a 90% success rate and convergence, our method requires the fewest steps for each task, efficiently transitioning from high performance to stabilization. Third, based on step statistics and the average time per step, our method takes slightly more training time than FERM for the Reach and Pick-and-Place tasks, while demonstrating clear time efficiency advantages over SVEA (CNN encoder) and SVEA (ViT encoder). Especially for the Push task, DCCT shows a significant time efficiency advantage. This efficiency highlights the superiority of lite ViT over the original in robotic manipulation. Forth, Table 1 indicates that the Push task is the most challenging, requiring the maximum training steps and resulting in the lowest final success rate. This difficulty arises because, unlike other tasks with straightforward trajectories, the Push task involves complex interactions between the block and the end-effector. Factors such as the shape of the block and end-effector, contact position, and physical parameters cause deviations from the intended trajectory, necessitating frequent trajectory adjustments during policy training, which is particularly challenging.

ISSN 2706-655X Vol.6, Issue 5: 67-78, DOI: 10.25236/IJFET.2024.060510

Table 1: Performance comparison in robotic manipulation

Tasks	Metrics	Reach	Pick-and-Place	Push	Average Time of Each Step (ms)
FERM	Steps to 90% success rate Steps to convergence Final success rate (%)	8K 19K 100	62K 78K 100	- 203K 65	32.2
SVEA (CNN encoder)	Steps to 90% success rate Steps to convergence Final success rate (%)	5K 10K 100	51K 72K 100	- 170K 78	102.3
SVEA (ViT encoder)	Steps to 90% success rate Steps to convergence Final success rate (%)	5K 12K 100	64K 83K 100	- 186K 81	128.6
Ours	Steps to 90% success rate Steps to convergence Final success rate (%)	5K 12K 100	62K 66K 100	95K 151K 94	91.4

4.3 Ablations

Building on the demonstrated efficiency of demonstrations and data augmentation in FERM, we focus on three key aspects of our method: the tokenization method for the Transformer section, the Transformer encoder, and unsupervised pretraining. These aspects help us examine the impact of the lite ViT-based image encoder on the training framework.

4.3.1 Patching Method for Transformer

To evaluate the impact of different methods for image patching in the Transformer section, we tested embedding images into patches using both CNN and standard segmentation techniques. We employed the original 16x16 patch size from ViT and a smaller 8x8 patch size, more suited for small datasets. After segmentation, we applied linear projection and reshaped the embeddings for the Transformer's encoding layer, adding positional embeddings to retain spatial information. The training success rates are illustrated in Figure 4. Our results show that the original ViT 16x16 patch size method fails to converge, while both the 8x8 patch size and the CNN encoder achieve high success rates. The poor performance of the original patch size method may be due to its lack of inductive bias and its high data requirements, which could be mitigated by increasing the number of pretraining images. However, ViT's data requirements are significantly higher than CNN's, making it less efficient. Replacing the CNN encoder improved convergence speed and performance compared to the smaller patch size method. These ablation results, along with FERM's performance, which uses only CNN as the image encoder and does not achieve outstanding results, highlight the effectiveness of combining Transformer and CNN-based patching methods in robotic manipulation training.

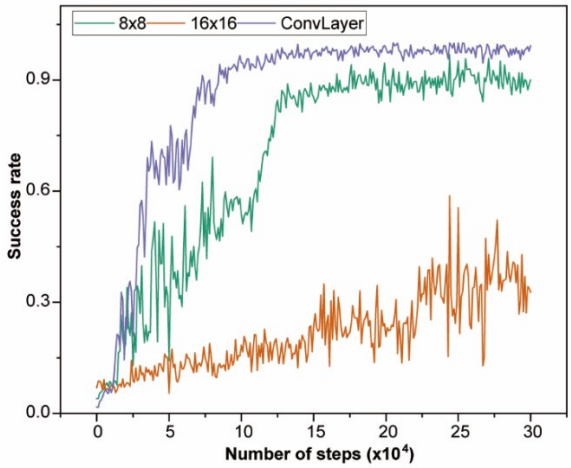


Figure 4: Performance of patching method ablation.

ISSN 2706-655X Vol.6, Issue 5: 67-78, DOI: 10.25236/IJFET.2024.060510

4.3.2 Transformer Encoder

We examined the impact of the Transformer section by comparing methods with and without the Transformer in the Reach task. To handle the Transformer's sequential token input format, format transition components were included in our method. In the ablation study, the Transformer was replaced with a single CNN part serving as the image encoder. The results, depicted in Figure 5 and referenced in Table 1, reveal that the Reach task, being relatively simple, allows the method with a CNN encoder to reach a high success rate. However, the convergence speed with the CNN encoder is somewhat slower than with FERM, and the steps to convergence in our encoder are nearly half that of the method using the CNN encoder. Without the Transformer's support, the framework's capability to learn features for policy optimization is weakened. These discrepancies in convergence speeds and performance become more significant with more challenging tasks, as detailed in Table 1.

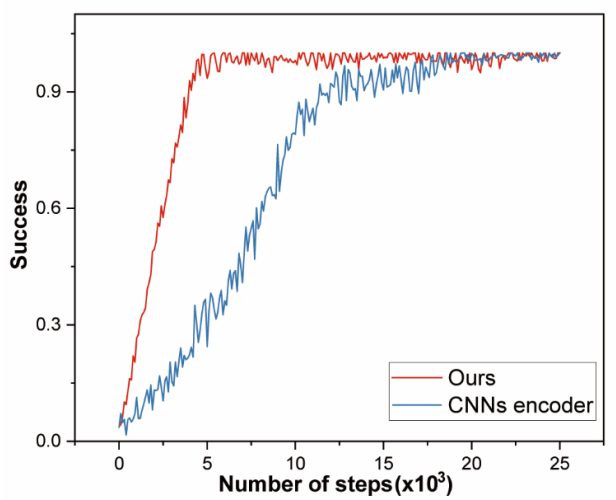


Figure 5: Performance of Transformer encoder ablation.

4.3.3 Unsupervised Pre-training

We investigated the impact of unsupervised pretraining with the Transformer by performing five experiments on the Pick-and-Place task, adjusting the number of pre-training iterations for encoder initialization to 0, 100, 1000, 2000, and 4000. The results, shown in Figure 6, indicate that the agent struggles to develop an optimal policy with 0 or 100 iterations. As the number of pre-training iterations increases, the agent's performance improves. Sufficient pre-training accelerates convergence during policy optimization and stabilizes the optimal policy. The performance improvement from 4000 to 2000 iterations is less significant compared to the improvement from 1000 to 2000 iterations, suggesting that 2000 iterations are sufficient for balancing learning efficiency and performance.

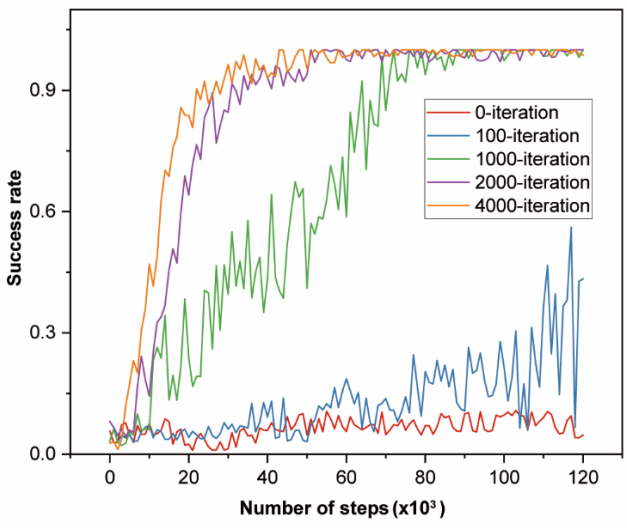


Figure 6: Performance of pre-training iterations ablation.

ISSN 2706-655X Vol.6, Issue 5: 67-78, DOI: 10.25236/IJFET.2024.060510

4.3.4 Saliency Maps

Saliency maps provide a visual interpretation of attention mechanisms. However, it is uncertain whether saliency maps are redundant when used as visual inputs for the Transformer's attention mechanism. To investigate the effect of saliency maps, we conducted ablation experiments on the visual inputs. The experimental results are shown in Figure 7. The combination of eye-in-hand and eye-to-hand visual inputs is sufficient for the reach and pick-and-place tasks. However, the inclusion of saliency maps helps the framework achieve a higher success rate in the push task, highlighting the advantages of saliency maps in robot visual manipulation tasks.

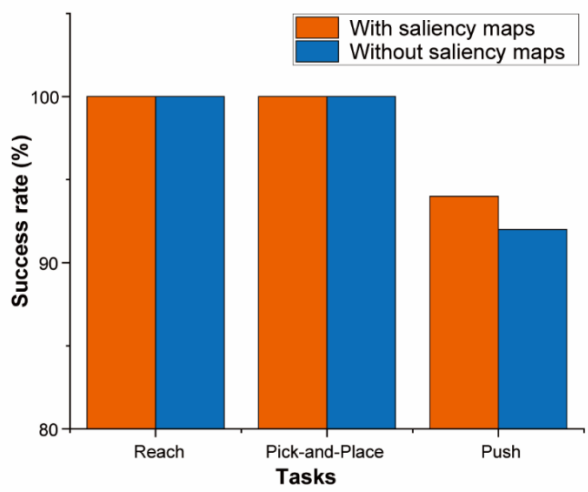


Figure 7: Performance of saliency ablation

5. Conclusion

The overall performance of vision-based efficient robotic manipulation across multiple tasks remains a persistent challenge. In this study, we have proposed an effective RL framework incorporating a novel vision encoding method based on lite ViT, and introduced saliency maps into the visual inputs. This significantly improves the general performance of three robotic manipulation tasks in the OpenAI simulated environment. Our method employs a three-channel encoder to synchronously extract and integrate visual information from different perspectives, enhancing visual robustness across different tasks. Additionally, we utilize unsupervised learning, data augmentation, and SAC to achieve data efficiency in the training process. Furthermore, our experiments demonstrate the importance of saliency maps in improving success rates for push tasks. This research is crucial for various application domains requiring robust multi-task performance with limited training data. Future work will focus on evaluating the general performance of our method with real-world robots, expanding the range of robotic manipulation tasks, and reducing demonstration time.

Acknowledgement

This work is partly funded by Technique Program of Jiangsu (No.BE2021086).

References

[1] Badia, A.P.; Piot, B.; Kapturowski, S.; Sprechmann, P.; Vitvitskyi, A.; Guo, Z.D.; Blundell, C. Agent57: Outperforming the atari human benchmark. In Proceedings of the International Conference on Machine Learning, Virtual, 13–18 July 2020; pp. 507–517.
[2] Berner, C.; Brockman, G.; Chan, B.; Cheung, V.; De¸biak, P.; Dennison, C.; Farhi, D.; Fischer, Q.; Hashme, S.; Hesse, C.; et al. Dota 2 with large scale deep reinforcement learning. arXiv 2019, arXiv:1912.06680.
[3] Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. Nature 2015, 518, 529–533.
[4] Vinyals, O.; Babuschkin, I.; Czarnecki, W.M.; Mathieu, M.; Dudzik, A.; Chung, J.; Choi, D.H.;

ISSN 2706-655X Vol.6, Issue 5: 67-78, DOI: 10.25236/IJFET.2024.060510

Powell, R.; Ewalds, T.; Georgiev, P.; et al. Grandmaster level in StarCraft II using multi-agent reinforcement learning. Nature 2019, 575, 350–354.
[5] Yang, Y.; Caluwaerts, K.; Iscen, A.; Zhang, T.; Tan, J.; Sindhwani, V. Data efficient reinforcement learning for legged robots. In Proceedings of the Conference on Robot Learning, Virtual, 16–18 November 2020; pp. 1–10.
[6] Haarnoja, T.; Ha, S.; Zhou, A.; Tan, J.; Tucker, G.; Levine, S. Learning to walk via deep reinforcement learning. arXiv 2018, arXiv:1812.11103.
[7] Jangir, R.; Hansen, N.; Ghosal, S.; Jain, M.; Wang, X. Look Closer: Bridging Egocentric and Third-Person Views with Transformers for Robotic Manipulation. IEEE Robot. Autom. Lett. 2022, 7, 3046–3053.
[8] Hansen, N.; Su, H.; Wang, X. Stabilizing deep q-learning with convnets and vision transformers under data augmentation. Adv. Neural Inf. Process. Syst. 2021, 34, 3680–3693.
[9] Chen, T.; Xu, J.; Agrawal, P. A system for general in-hand object re-orientation. In Proceedings of the Conference on Robot Learning, Auckland, New Zealand, 8–11 November 2022; pp. 297–307.
[10] Johannink, T.; Bahl, S.; Nair, A.; Luo, J.; Kumar, A.; Loskyll, M.; Ojea, J.A.; Solowjow, E.; Levine, S. Residual Reinforcement Learning for Robot Control. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 6023–6029. https://doi.org/10.1109/ICRA.2019.8794127.
[11] Andrychowicz, O.M.; Baker, B.; Chociej, M.; Jozefowicz, R.; McGrew, B.; Pachocki, J.; Petron, A.; Plappert, M.; Powell, G.; Ray, A.; et al. Learning dexterous in-hand manipulation. Int. J. Robot. Res. 2020, 39, 3–20.
[12] Levine, S.; Pastor, P.; Krizhevsky, A.; Ibarz, J.; Quillen, D. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. Int. J. Robot. Res. 2018, 37, 421–436.
[13] OpenAI, O.; Plappert, M.; Sampedro, R.; Xu, T.; Akkaya, I.; Kosaraju, V.; Welinder, P.; D'Sa, R.; Petron, A.; Pinto, H.P.d.O.; et al. Asymmetric self-play for automatic goal discovery in robotic manipulation. arXiv 2021, arXiv:2101.04882.
[14] Gu, S.; Holly, E.; Lillicrap, T.; Levine, S. Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–03 June 2017, pp. 3389–3396.
[15] Peng, X.B.; Andrychowicz, M.; Zaremba, W.; Abbeel, P. Sim-to-real transfer of robotic control with dynamics randomization. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018; pp. 3803–3810.
[16] Yu, W.; Tan, J.; Bai, Y.; Coumans, E.; Ha, S. Learning fast adaptation with meta strategy optimization. IEEE Robot. Autom. Lett. 2020, 5, 2950–2957.
[17] James, S.; Bloesch, M.; Davison, A.J. Task-embedded control networks for few-shot imitation learning. In Proceedings of the Conference on Robot Learning, Zürich, Switzerland, 29–31 October 2018; pp. 783–795.
[18] Finn, C.; Yu, T.; Zhang, T.; Abbeel, P.; Levine, S. One-shot visual imitation learning via meta-learning. In Proceedings of the Conference on Robot Learning, Mountain View, CA, USA, 13–15 November 2017; pp. 357–368.
[19] Duan, Y.; Andrychowicz, M.; Stadie, B.; Jonathan Ho, O.; Schneider, J.; Sutskever, I.; Abbeel, P.; Zaremba, W. One-shot imitation learning. Adv. Neural Inf. Process. Syst. 2017, 30. https://doi.org/10.48550/arXiv.1703.07326.
[20] Wu, Y.H.; Charoenphakdee, N.; Bao, H.; Tangkaratt, V.; Sugiyama, M. Imitation learning from imperfect demonstration. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 09–15 June 2019; pp. 6818–6827.
[21] Laskin, M.; Srinivas, A.; Abbeel, P. Curl: Contrastive unsupervised representations for reinforcement learning. In Proceedings of the International Conference on Machine Learning, Virtual, 13–18 July 2020; pp. 5639–5650.
[22] Laskin, M.; Lee, K.; Stooke, A.; Pinto, L.; Abbeel, P.; Srinivas, A. Reinforcement learning with augmented data. Adv. Neural Inf. Process. Syst. 2020, 33, 19884–19895.
[23] Yarats, D.; Zhang, A.; Kostrikov, I.; Amos, B.; Pineau, J.; Fergus, R. Improving sample efficiency in model-free reinforcement learning from images. arXiv 2019, arXiv:1910.01741.
[24] Zhan, A.; Zhao, P.; Pinto, L.; Abbeel, P.; Laskin, M. A framework for efficient robotic manipulation. arXiv 2020, arXiv:2012.07975.
[25] Hassani, A.; Walton, S.; Shah, N.; Abuduweili, A.; Li, J.; Shi, H. Escaping the big data paradigm with compact transformers. arXiv 2021, arXiv:2104.05704.
[26] Wu, Z.; Liu, Z.; Lin, J.; Lin, Y.; Han, S. Lite transformer with long-short range attention. arXiv 2020, arXiv:2004.11886.

ISSN 2706-655X Vol.6, Issue 5: 67-78, DOI: 10.25236/IJFET.2024.060510

[27] Wang, H.; Wu, Z.; Liu, Z.; Cai, H.; Zhu, L.; Gan, C.; Han, S. Hat: Hardware-aware transformers for efficient natural language processing. arXiv 2020, arXiv:2005.14187.
[28] Wu, B.; Xu, C.; Dai, X.; Wan, A.; Zhang, P.; Yan, Z.; Tomizuka, M.; Gonzalez, J.; Keutzer, K.; Vajda, P. Visual transformers: Token-based image representation and processing for computer vision. arXiv 2020, arXiv:2006.03677.
[29] Graham, B.; El-Nouby, A.; Touvron, H.; Stock, P.; Joulin, A.; Jégou, H.; Douze, M. LeViT: A Vision Transformer in ConvNet's Clothing for Faster Inference. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 12259–12269.
[30] Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; Jégou, H. Training data-efficient image transformers & distillation through attention. In Proceedings of the International Conference on Machine Learning, Virtual, 18–24 July 2021; pp. 10347–10357.
[31] Krizhevsky, A., & Hinton, G. (2009). Learning multiple layers of features from tiny images. http://www.cs.utoronto.ca/~kriz/learning-features-2009-TR.pdf.
[32] Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arxiv preprint arxiv:1910.01108.
[33] Schwarzer, M.; Anand, A.; Goel, R.; Hjelm, R.D.; Courville, A.; Bachman, P. Data-efficient reinforcement learning with self- predictive representations. arXiv 2020, arXiv:2007.05929.
[34] Jaderberg, M.; Mnih, V.; Czarnecki, W.M.; Schaul, T.; Leibo, J.Z.; Silver, D.; Kavukcuoglu, K. Reinforcement learning with unsupervised auxiliary tasks. arXiv 2016, arXiv:1611.05397.
[35] Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2019). Albert: A lite bert for self-supervised learning of language representations. arxiv preprint arxiv:1909.11942.
[36] Koonce, B., & Koonce, B. E. (2021). Convolutional neural networks with swift for tensorflow: Image recognition and dataset categorization. New York, NY, USA: Apress, 109-123.
[37] Tan, M., Pang, R., & Le, Q. V. (2020). Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 10781-10790).
[38] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. arxiv preprint arxiv:2010.11929.
*[39] He, K., Fan, H., Wu, Y., **e, S., & Girshick, R. (2020). Momentum contrast for unsupervised visual representation learning. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 9729-9738).*
[40] Chen, L.; Lu, K.; Rajeswaran, A.; Lee, K.; Grover, A.; Laskin, M.; Abbeel, P.; Srinivas, A.; Mordatch, I. Decision transformer: Reinforcement learning via sequence modeling. Adv. Neural Inf. Process. Syst. 2021, 34, 15084–15097.
[41] Chen, C.; Wu, Y.F.; Yoon, J.; Ahn, S. Transdreamer: Reinforcement learning with transformer world models. arXiv 2022, arXiv:2202.09481.
[42] Tao, T.; Reda, D.; van de Panne, M. Evaluating Vision Transformer Methods for Deep Reinforcement Learning from Pixels. arXiv 2022, arXiv:2204.04905.
[43] Meng, L.; Goodwin, M.; Yazidi, A.; Engelstad, P. Deep Reinforcement Learning with Swin Transformer. arXiv 2022, arXiv:2206.15269.
[44] Li, G.; Lin, S.; Li, S.; Qu, X. Learning Automated Driving in Complex Intersection Scenarios Based on Camera Sensors: A Deep Reinforcement Learning Approach. IEEE Sens. J. 2022, 22, 4687–4696.
[45] Henaff, O. Data-efficient image recognition with contrastive predictive coding. In Proceedings of the International Conference on Machine Learning, Virtual, 13–18 July 2020; pp. 4182–4192.
[46] Chen, T.; Kornblith, S.; Norouzi, M.; Hinton, G. A simple framework for contrastive learning of visual representations. In Proceedings of the International Conference on Machine Learning, Virtual, 13–18 July 2020; pp. 1597–1607.
[47] Grill, J. B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., ... & Valko, M. (2020). Bootstrap your own latent-a new approach to self-supervised learning. Advances in neural information processing systems, 33, 21271-21284.
[48] Xie, Q.; Dai, Z.; Hovy, E.; Luong, T.; Le, Q. Unsupervised data augmentation for consistency training. Adv. Neural Inf. Process. Syst. 2020, 33, 6256–6268.
[49] Berthelot, D.; Carlini, N.; Goodfellow, I.; Papernot, N.; Oliver, A.; Raffel, C.A. Mixmatch: A holistic approach to semi-supervised learning. Adv. Neural Inf. Process. Syst. 2019, 32. https://doi.org/10.48550/arXiv.2001.07685.
[50] Sohn, K.; Berthelot, D.; Carlini, N.; Zhang, Z.; Zhang, H.; Raffel, C.A.; Cubuk, E.D.; Kurakin, A.; Li, C.L. FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence. Adv. Neural Inf. Process. Syst. 2020, 33, 596–608.
[51] Van den Oord, A.; Li, Y.; Vinyals, O. Representation learning with contrastive predictive coding. arXiv 2018, arXiv:1807.03748.