stroM E: Redshift Prediction Using a Masked utoencoder with a Novel Fine-Tuning rchitecture

mirreza Dolatpour Fathkouhi
Department of Computer Science
University of Virginia
Charlottesville, US
aww9gh@virginia.edu

Geoffrey Charles Fox
Department of Computer Science
University of Virginia
Charlottesville, US
vxj6mb@virginia.edu

***bstract*—Redshift prediction is a fundamental task in astronomy, essential for understanding the expansion of the universe and determining the distances of astronomical objects. ccurate redshift prediction plays a crucial role in advancing our knowledge of the cosmos. Machine learning (ML) methods, renowned for their precision and speed, offer promising solutions for this complex task. However, traditional ML algorithms heavily depend on labeled data and task-specific feature extraction. To overcome these limitations, we introduce stroM E, an innovative approach that pretrains a vision transformer encoder using a masked autoencoder method on Sloan Digital Sky Survey (SDSS) images. This technique enables the encoder to capture the global patterns within the data without relying on labels. To the best of our knowledge, stroM E represents the first application of a masked autoencoder to astronomical data. By ignoring labels during the pretraining phase, the encoder gathers a general understanding of the data. The pretrained encoder is subsequently fine-tuned within a specialized architecture tailored for redshift prediction. We evaluate our model against various vision transformer architectures and CNN-based models, demonstrating the superior performance of stroM E's pretrained model and fine-tuning architecture.**

***Index Terms*—Masked autoencoder, Redshift prediction, SDSS, Self-supervised learning, Fine-tuning, Deep learning**

I. Introduction

Redshift prediction is one of the most compelling areas of study in astronomy, offering insights into the universe's expansion and the distances of celestial objects such as quasars, stars, and galaxies [1]. ccurately capturing spectral features over extended periods is crucial for redshift prediction, but this is feasible for only 1% of galaxies. dditionally, most telescopes can simultaneously capture spectra from only a limited number of objects [2]. Consequently, photometric methods have been proposed as alternatives, since spectroscopic methods are both expensive and time-consuming [3]. Thanks to advancements in telescopes, a vast number of images are captured and made available by various surveys, such as DESI [4] and Hyper Suprime-Cam [5].

Photometric redshift prediction can be achieved through two main approaches: template-fitting methods and machine learning-based methods, particularly deep learning. Template-fitting methods, such as those researched by Salvato et al. [6], aim to determine the probability density function of redshift [7]. Given this paper's focus on deep learning, most reviews explore deep learning-based methods in astronomy, specifically for redshift prediction.

Deep learning methods proposed for astronomy can mainly be categorized into supervised and self-supervised learning algorithms.

In supervised learning, the model's training process is dependent on the availability of labeled data, where the target values guide the learning of task-specific features. Dey et al. [2] proposed a method based on capsule cells. Pasquet et al. [9] used an Inception model to extract features from images, which were then concatenated with galactic reddening values. Rastegarnia et al. [10] used residual blocks to predict quasar redshifts. In [11], a deep learning model based on a convolutional neural network (CNN) was designed to predict quasar redshifts. Sandeep et al. [12], in addition to using pretrained models such as lexNet, VGG16, and ResNet50, proposed a new CNN-based model to classify galaxies and predict their redshifts. Syarifudin et al. [13] used multi-band images and a DenseNet model to predict redshifts. Schuldt et al. [14] proposed Netz, a CNN deep learning model trained on five-filter images collected from the Hyper Suprime-Cam Subaru Strategic Program. In [15], a Vision Transformer (ViT) model, based on the transformer architecture, was trained for galaxy classification.

Supervised learning methods require labels for training, and the extracted features are specifically related to the defined task. dditionally, labeled data is not widely available, and the extracted features often fail to capture the general patterns of the data. To address this, self-supervised learning methods have been proposed to leverage abundant data without relying on labels. These methods define a pretext task [16] that the model solves, such as image inpainting [17], allowing the model to learn general patterns and features of the data.

Self-supervised learning includes two main phases. The first phase is pretraining, where the model is trained on the pretext task using unlabeled data to identify general patterns. The second phase is fine-tuning, where the pretrained model is used to solve a specific task. Hayat et al. [18] used contrastive learning for pretraining the ResNet50 model, with the pretrained weights later employed for morphology classification and redshift prediction. Lanusse et al. [19] proposed stroCLIP, a multimodal model pretrained using a contrastive learning

approach on the DESI survey, which was then used for redshift and stellar mass prediction. Shen et al. [20] pretrained ResNet50 with a momentum contrastive learning method, later employing the pretrained model for Galaxy Zoo classification. Stein et al. [21] pretrained ResNet50 for similarity search between galaxies. Oliva et al. [22] pretrained a transformer-based architecture using a masking strategy on millions of R-band light curves.

Previous studies mainly used various contrastive learning methods [23] for pretraining. These methods are highly sensitive to the selected augmentation techniques [24], [25]. Incorrect augmentations can significantly degrade model performance. Furthermore, with a greater number of different views, more patches are processed by the encoder. In contrast, Masked utoEncoder (M E) [26], a self-supervised learning method, uses only 25% of patches, making it more efficient compared to contrastive learning methods. ccording to experiments conducted in [26], M E is not sensitive to the type of augmentation techniques used. M E employs a vision transformer [27] as the encoder, which can extract the global dependency and general patterns of images but lacks the ability to capture locality information, such as edges, which CNNs can intuitively extract.

To address the aforementioned issues, this paper makes the following contributions:

- We pretrained a model on a portion of the SDSS survey images by ignoring the labels, using a masked autoencoder approach to gather the general and global features of the data. This method aims to achieve faster, more efficient pretraining that is less sensitive to augmentation. To the best of our knowledge, this is the first paper to explore the usage of masked autoencoders for astronomy images.
- To mitigate the lack of locality in vision transformer models, we introduce a novel fine-tuning method specifically designed for redshift prediction.
- We design various architectures based on vision transformers and CNNs to address these issues and demonstrate the superiority of our proposed stroM E through different experiments.

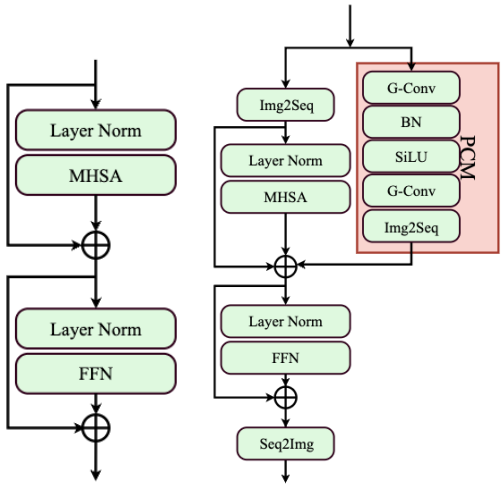
The remainder of this paper is structured as follows: Section II delves into the stroM E architecture and outlines the proposed fine-tuning methods. Section III presents our experiments, detailing both the pretraining and fine-tuning processes, along with a comprehensive analysis of our results. Ultimately, we conclude with a summary of our findings and the implications of our experiments.

II. Proposed Method

This section begins with an introduction to the architectures and concepts utilized in this paper. Then, the proposed model architectures are presented.

. *Vision Transformer*

The Vision Transformer (ViT), proposed by Dosovitskiy et al. [27], aims to train images using a plain transformer layer.



(a) (b)

Fig. 1: These diagrams illustrate (a) a plain-transformer and (b) a pcm-transformer. The abbreviations used are as follows: Layer Norm (Layer Normalization), MHS (Multi-Head Self-ttention), FFN (Feed Forward Network), and BN (Batch Normalization). Img2Seq and Seq2Img represent the processes of converting between 1D and 2D features. G-Conv denotes a group of convolutional layers, and SiLU layer is explained in [28].

It begins by segmenting an image $x \in R^{H \times W \times}$ into uniform patches $x_t \in R^{\frac{W)}{p^2} \times D}$, where H, W, and C represent the height, width, and channel of the image x, and p represents the size of each patch, respectively. These patches are then transformed into embedding vectors of size $D = p^2C$ via a linear projector. learnable class token is also concatenated with these patch embedding vectors. Subsequently, positional embedding vectors are added pairwise to the patch embeddings to inform the transformer layers about their positions in the image.

The plain transformer layer includes multi-head self-attention (MHS) and a feed-forward network (FFN).

MHS : Each token x_t is projected to query (Q), key (K),and Value (V) vectors for *h* times using $W_Q, W_K, W_V \in R^{D \times D}$.

$$Q = W_Q \times x_t \tag{1}$$

$$K = W_K \times x_t \tag{2}$$

$$V = W_V \times x_t \tag{3}$$

For each of these projections, self-attention is conducted as demonstrated below:

$$Attention(Q, K, V) = Softmax(\frac{QK^T}{\sqrt{D}})V \tag{4}$$

The output of the MHS is the concatenation of all calculated *ttention* outputs along the channel dimension.

FFN: This module contains two linear layers with a GeLU activation function [29] added between them.

s shown in Fig. 1, in each plain transformer layer, the input is normalized before being fed to each module, and a residual connection is added afterward.

B. Transformer Layer with Parallel Convolution Module

One drawback of the Vision Transformer (ViT) is that it cannot capture the local information of the image, which could be intuitively obtained by a Convolutional Neural Network (CNN). In ViT, images are treated as sequences of 1D tokens, and the 2D structure of images is not retained during training. In contrast, CNN applies kernel operators directly to images, effectively capturing the correlation between pixels. The Multi-Head Self ttention (MHS) module of the plain transformer can model global dependencies by calculating the attention, but only the Feed-Forward Network (FFN) module is dedicated to capturing local information [27].

To address this issue, a modified version of ViT E [30] leverages both CNN and plain transformer layers [31]. In this method, a parallel convolution-based module (PCM) is utilized and integrated with the output of the attention module. Consequently, the inductive bias obtained by the CNN is combined with the global dependency captured by the MHS . Fig. 1 illustrates the modified transformer layer proposed in [31]. In this paper, we refer to the modified transformer as the pcm-transformer.

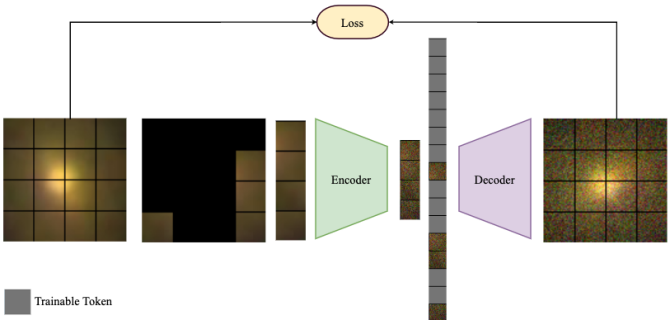


Fig. 2: It illustrates the architecture of pretraining the stro-M E using a masked autoencoder algorithm. Loss involves comparing the generated patches corresponding to masked areas with their original counterparts.

C. Masked utoEncoder

The Masked utoencoder (M E), a pre-training technique developed by Facebook [26], is an asymmetrical autoencoder designed to extract patterns from data by reconstructing original images from portions of those images. The M E comprises an encoder and a decoder. The encoder is a Vision Transformer (ViT) that features transformer-based layers, while the decoder can be constructed using either transformer or linear layers [32].

The process begins with segmenting images into uniform patches, which are subsequently transformed into embedding vectors via a convolution-based patch embedder. Positional embeddings—computed using sine-cosine functions—are integrated into the patch embeddings to inform the encoder about the position of each patch. predetermined mask ratio dictates the random removal of certain patches, with the remaining patches then fed into the encoder. To make the problem more challenging for the M E and avoid using extrapolation to predict masked patches from neighboring pixels, a high masking ratio is typically used (75%). In place of the masked patches, learnable tokens are generated and combined with the output of the encoder. These tokens are further enhanced with positional embeddings before being forwarded to the decoder. Ultimately, the decoder attempts to reconstruct the original image, and the model's effectiveness is assessed based on how accurately the reconstructed patches match their corresponding original segments. By masking patches and feeding only a small fraction of them, the pre-training process becomes significantly faster and more efficient. The M E architecture is illustrated in Fig. 2.

D. Inception module

The Inception module [33] is constructed using CNN. In this architecture, four parallel branches of convolutions are integrated to increase both the depth and the width of the model. To make the process more efficient and decrease computations, a convolution layer with a kernel size of 1×1 is added before convolutions with sizes 3×3 and 5×5. The Inception module architecture is shown in Fig. 3.

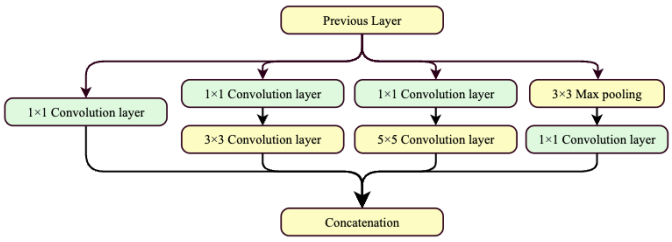


Fig. 3: Inception module

E. stroM E (Proposed Method)

stroM E, involves using a masked autoencoder for pretraining and utilizing the pretrained encoder in a novel fine-tuning architecture designed for redshift prediction.

Pretraining: stroM E is pretrained using a masked autoencoder. We employed both plain-transformer and pcm-transformer layers for constructing the stroM E. We pretrained two versions of stroM E: one based on the plain-transformer layer, called plain- stroM E, and another using pcm-transformer layers, called pcm- stroM E.

Fine-tuning: The proposed fine-tuning model, depicted in Fig. 7, contains three separate modules explained below:

a) Pretrained Encoder: The decoder part of the stro-M E is discarded, and only the encoder is used for fine-tuning. Two linear layers with a ReLU activation function between them serve as the head of the encoder. dditionally, fine-tuning is done partially, meaning that all weights of the pretrained encoder, except for those in the head, are frozen.

b) Inception Model: This branch of the architecture contains five inception blocks, as explained earlier. The first four blocks include all four parallel branch convolution layers. However, in the last inception block, the branch containing the convolution layer with a kernel size of 5×5 is omitted. This is because the input to this branch is too small to apply a convolution with a 5×5 kernel.

c) Magnitude Block: This block includes a multi-layer perceptron comprising five linear layers with ReLU activation functions between them.

s depicted in Fig. 7, the proposed fine-tuning architecture consists of the concatenation of the Inception model, the magnitude block, and the frozen pretrained encoder, which are then fed into two linear layers with a ReLU activation function between them.

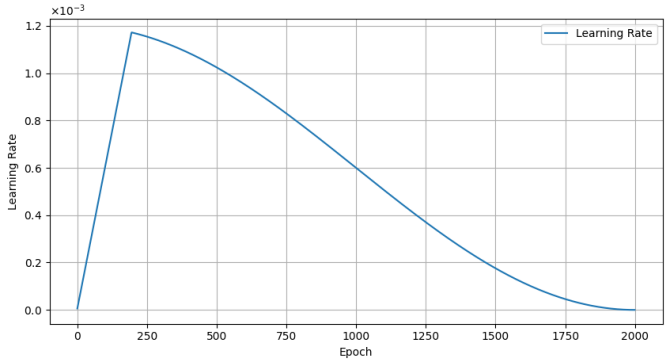


Fig. 4: Learning rate during pretraining.

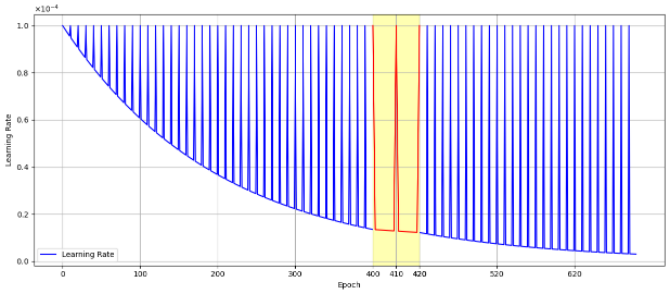


Fig. 5: It shows the learning rate during fine-tuning, with the yellow section highlighting the changes over two cycles.

T BLE I: Fine-tuning Hyperparameters

Hyperparameter	Value
$lr_{initi\ l}$	1e-4
batch size	1,024
seed	42
total epochs	700
optimizer	damW [8]
weight decay	0.005
betas	(0.9, 0.999)

T BLE II: Pretraining Hyperparameters

Hyperparameter	Value
$lr_{pe\ k}$	1.17e-3
batch size	2,048
mask ratio	0.75
seed	42
total epochs	2,000
optimizer	damW
$epo\ h_{w\ rm\ up}$	196
weight decay	0.05
betas	(0.9, 0.95)

III. Experiment l Results

We conducted two experimental setups, utilizing 80% and 100% of the image dataset for pretraining in the first and second experiments, respectively. In the second experiment, we compared the best-performing stroM E architecture from the first experiment, pcm- stroM E, with the baseline method proposed by Henghes et al. [34]. This comparison was conducted to assess the generality and robustness of the proposed methods when applied to a larger dataset, ensuring that the performance gains observed with 80% of the data scale effectively with the full dataset.

Implementation: PyTorch, one of the well-known implementation frameworks, was utilized for this study. Special thanks to Rivanna High-Performance Computing for providing the necessary computational resources. Pretraining was conducted using four 100 GPUs, typically taking two to three days to complete. Fine-tuning experiments were performed based on GPU availability, utilizing either one or four 100 GPUs. The initial fine-tuning took approximately 1 hour using one GPU, while fine-tuning the full set of labeled data typically took around 10 hours when using four GPUs.

Dataset: The dataset provided by Pasquet et al. [9] contains 659,857 images with 64 corresponding physical properties. Physical properties, such as spectroscopic redshift z are collected from the 12th version of the Sloan Digital Sky Survey (SDSS DR12) [35]–[37]. Images corresponding to these physical properties are retrieved from the DR8 SDSS survey. The images contain five bands, including u, g, r, i, and z frames, and the size of each image is $64 \times 64 \times 5$. ll raw images are preprocessed by background subtraction and the same zero-point photometric calibration. More information related to the dataset and preprocessing steps is explained in [9].

. *First Experiment:*

Pretraining Data: s mentioned before, only images are utilized for pretraining. Therefore, we ignored around 80% of the labels from the complete dataset to gather the global dependencies of the data and capture non-specific patterns. Consequently, 527,886 images are dedicated to the training

data. To monitor the behavior of the model during pretraining, approximately 10% of the whole dataset is set aside for validation. Similar to the training data, the labels of the validation set are ignored.

Fine-tuning: The fine-tuning data consists of images along with their corresponding magnitude values, including u, g, r, i, and z, in addition to spectroscopic redshift z as the target, representing 10% of the entire dataset. Moreover, the u, g, r, i, and z magnitude values are obtained using the astroquery library [38]. The data distribution for training, validation, and testing comprises 70%, 10%, and 20% of the fine-tuning data, respectively. The fine-tuning data is also used for models trained from scratch, as shown in IV.

s mentioned in [34], there is no significant difference between the results for images of size $32 \times 32 \times 5$ and those of size $64 \times 64 \times 5$. Based on this, images are cropped from the center to a size of $32 \times 32 \times 5$ during both pretraining and fine-tuning. To increase the difficulty and prevent overfitting, random rotation at 45 degrees, along with horizontal and vertical flipping methods, are applied to the training images during both the pretraining and fine-tuning phases. dditionally, Gaussian noise with a standard deviation of 0.05 is used during fine-tuning.

T BLE III: stroM E Pretraining rchitectures

Parameter	Component	Value
patch size	Encoder	8
	Decoder	8
embedding size	Encoder	192
	Decoder	192
depth	Encoder	12
	Decoder	4
number of heads	Encoder	3
	Decoder	3

1) Training Configurations:

Learning Rate Scheduler: In previous deep learning training, models were trained using a constant learning rate. This method can cause the model to underperform and is not effective for optimizing deep models. On this account, two schedulers are employed for both pretraining and fine-tuning. Both pretraining and fine-tuning learning rates lr are demonstrated in Fig. 4, 5. dditionally, optimization details are mentioned in Table I, II.

a) Pretraining: Similar to [39], a Cosine nnealing with Warm-Up scheduler is used for pretraining. This scheduler consists of two phases. In the first phase, the learning rate increases linearly from a low value to a high learning rate, lr_{peak}, over a specific number of epochs, called $epoch_{warm\ \ up}$. In the second phase, the scheduler decreases the learning rate using cosine decay. Linearly increasing the learning rate avoids unstable training and improves the global search of the optimizer, while the cosine decay decreases the learning rate more smoothly, providing a good balance between global and local search.

b) Fine-tuning: cyclic scheduler is employed that restarts the lr every 10 epochs, then decreases exponentially using $(0\ 995)^{epoch}$. This strategy helps the model escape local minima and achieve good convergence in the final epochs.

stroM E Hyperparameters: For pretraining, compared to other papers that utilized M E, our data is limited and not as large. Based on the analysis conducted in [40], larger models require training on larger datasets for a higher number of epochs. Therefore, we built small models by setting configurations mentioned in Table III for both plain- stroM E and pcm- stroM E.

For fine-tuning, it is important to note that the encoder cannot be used directly as some shuffling is applied to patches during M E pretraining. Consequently, the weights of the encoder should be extracted and used to initialize a new ViT model for fine-tuning. During fine-tuning, all layers except the last two layers of the ViT encoder (the layer normalization [41] and projection layer) are frozen.

B. Compare with other redshift predition methods

To demostrate the superiority of stroM E and the proposed fine-tuning architecture, we compared them with other redshift prediction model, which is based on the vision transformers or CNNs. ll architectures are illustrated in Fig. 6, 7.

plain-ViT and pcm-ViT: The pretrained encoders of plain-stroM E and pcm- stroM E are employed for fine-tuning. lightweight trainable head module is added at the end of the encoders for redshift prediction. In these architectures, the pretrained encoders are frozen during fine-tuning.

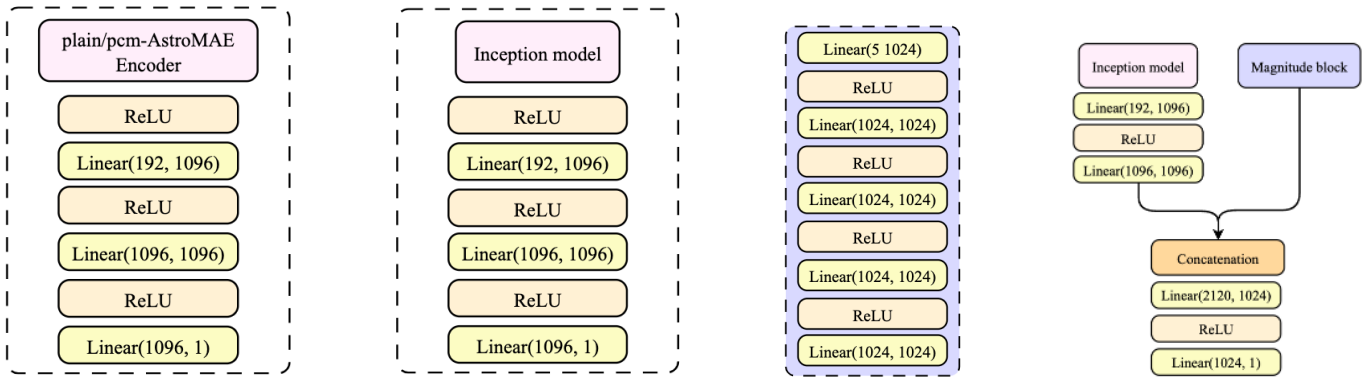
from-scratch plain-ViT and from-scratch pcm-ViT: The architectures are the same as plain-ViT and pcm-ViT, but the encoders are initialized randomly and are trainable during training. dditionally, no pretraining is conducted on these architectures.

Inception-only redshift prediction: Similar to the Inception model discussed in the fine-tuning model architecture, it includes five inception modules, with the last one lacking the 5×5 convolution layer. Finally, three linear layers with ReLU activations between them are added. ll weights of this model are trainable.

Henghes et al. [34]: In this paper, the Inception model is concatenated with the magnitude blocks. The output is then fed to two linear layers, with a ReLU activation function inserted between them.

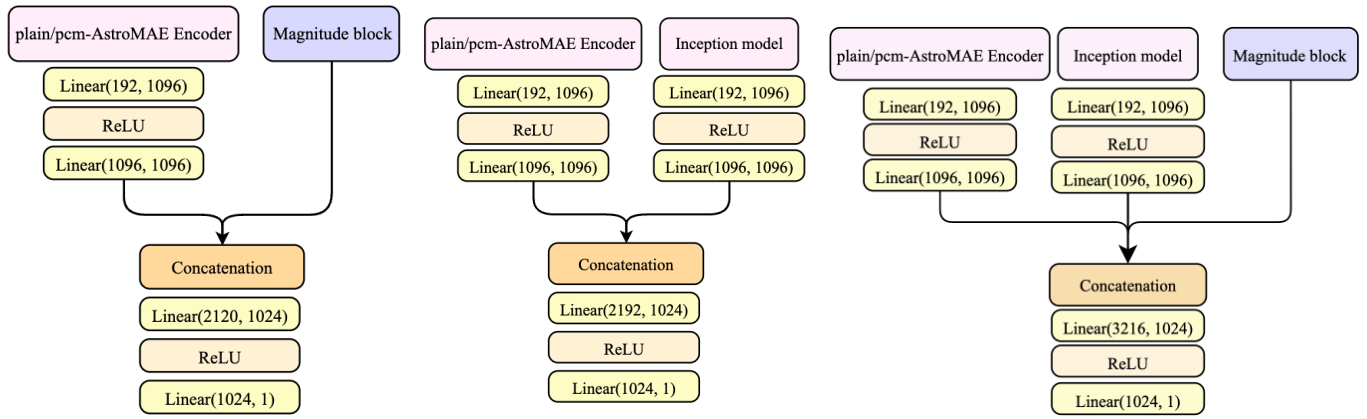
plain-ViT-inception and pcm-ViT-inception: Plain-ViT and pcm-ViT are concatenated with the Inception model. Two linear layers with one ReLU function are then used for redshift prediction. Except for plain-ViT and pcm-ViT, all weights of the models are trainable.

plain-ViT-magnitude and pcm-ViT-magnitude: The magnitude block output is concatenated with the plain-ViT or pcm-ViT output before being fed to the linear layers.



(a) (b) (c) (d)

Fig. 6: (a) plain-ViT, pcm-ViT, from-scratch plain-ViT, and from-scratch pcm-ViT, (b) Inception-only redshift prediction, (c) Magnitude Block, and (d) Henghes et al. [34] model.



(a) (b) (c)

Fig. 7: (a) plain-ViT-magnitude, pcm-ViT-magnitude, from-scratch plain-ViT-magnitude, and from-scratch pcm-ViT-magnitude, (b) plain-ViT-inception and pcm-ViT-inception, (c) Proposed stroM E Fine-tuning rchitecture.

from-scratch plain-ViT-magnitude and from-scratch pcm-ViT-magnitude: In this architecture, the magnitude block is only concatenated with the output of from-scratch plain-ViT and from-scratch pcm-ViT.

C. Metrics

Five metrics are considered for evaluating and comparing our proposed methods with others. These metrics are explained below:

Mean Square Error (MSE): The average of the squared differences between the spectroscopic and predicted redshift values is calculated.

$$\text{MSE} = \frac{1}{n}\sum_{i=}^{n}(z_i^s \quad \hat{z}_i^s)^2 \tag{5}$$

Mean bsolute Error (M E): The absolute differences between the predicted and ground-truth spectroscopic redshifts are averaged.

$$\text{M E} = \frac{1}{n}\sum_{i=}^{n}|z_i^s \quad \hat{z}_i^s| \tag{6}$$

Bias: It measures the average of the residuals, as defined in [42].

$$\text{Bias} = \langle\frac{\hat{z}^s \quad z^s}{1+z^s}\rangle \tag{7}$$

Precision: s mentioned in [43], it measures the expected scatter.

$$\text{Precision} = 1\ 48 \times \text{median}(|\frac{\hat{z}^s \quad z^s}{1+z^s}|) \tag{8}$$

score: It evaluates how well a regression model predicts. The R^2 score lies between 0 and 1, and the closer the score is to 1, the better the model predicts.

$$R^2(z^s, \hat{z}^s) = 1 \quad \frac{\sum_{i=}^{n}(z_i^s \quad \hat{z}_i^s)^2}{\sum_{i=}^{n}(z_i^s \quad z^s)^2} \tag{9}$$

T BLE IV: Redshift Prediction Using Various rchitectures Based on Transformer Layers and CNNs

rchitectures		Metrics				
Type	**Name**	**MSE**	**M E**	**Bias**	**Precision**	**R^2**
Supervised training (from scratch)	from-scratch plain-ViT-magnitude	0.00077	0.01871	0.00153	0.01736	0.93580
	from-scratch pcm-ViT-magnitude	0.00057	0.01604	-0.00035	0.01458	0.95204
	Henghes et al. [34]	0.00058	0.01568	0.00108	0.01443	0.95176
	from-scratch plain-ViT	0.00097	0.02123	0.00049	0.01957	0.91871
	from-scratch pcm-ViT	0.00063	0.01686	-0.00122	0.01554	0.94764
	Inception-only redshift prediction	0.00064	0.01705	0.00132	0.01593	0.94625
Fine-tuning	plain-ViT-magnitude	0.00068	0.01740	**-0.00007**	0.01596	0.94334
	pcm-ViT-magnitude	0.00060	0.01655	-0.00095	0.01522	0.94939
	Proposed plain- stroM E	0.00056	0.01558	0.00097	0.01429	0.95336
	Proposed pcm- stroM E	**0.00053**	**0.01520**	-0.00037	**0.01391**	**0.95601**
	plain-ViT	0.00086	0.01970	-0.00060	0.01775	0.92790
	pcm-ViT	0.00084	0.01945	-0.00114	0.01737	0.92950
	plain-ViT-inception	0.00059	0.01622	-0.00009	0.01496	0.95029
	pcm-ViT-inception	0.00059	0.01601	0.00042	0.01458	0.95095

In the above formulas, z^s, $\hat{z}^s$, and z^s represent the ground-truth spectroscopic redshift, predicted redshift, and average value of the spectroscopic redshift, respectively. Moreover, n is the number of data samples. It is worth noting that methods with lower MSE, M E, Bias, and Precision, and higher R^2 indicate better results.

D. Result nalysis

In this section, we analyze the results and discuss the potential advantages and drawbacks of our approach. The performance metrics are summarized in Table IV. To further evaluate the performance of the predicted redshifts compared to their corresponding spectroscopic ground truths, we generated density scatter plots for all experiments. These plots are displayed in Fig. 9, 10, 11.

Masked autoencoder provides valuable information for fine-tuning through unlabeled images: s mentioned before, one reason behind pretraining is to extract general patterns from the data, which are not specifically associated with a single task. ccording to Table IV, the results of plain-ViT are much better compared to its from-scratch counterpart, as it has lower MSE, M E, and Precision, and higher R^2. This clearly demonstrates the power of the pretrained encoder of stroM E in identifying valuable general patterns.

pcm-transformer can improve the lack of locality in plain-transformer: pcm-ViT obtained better results in terms of most metrics compared to plain-ViT. However, the improvement is not as significant compared to the results obtained by from-scratch pcm-ViT versus from-scratch plain-ViT. This demonstrates that the PCM module can gather more local information related to redshift prediction in supervised learning compared to during pretraining.

Inception-only redshift prediction is still more powerful than vision transformer models: Results demonstrate that the Inception-only redshift prediction can extract more relevant features for redshift prediction. Based on research conducted by Si et al. [44], Inception modules can provide local information very well, including local edges and texture. This experiment shows that for redshift prediction, local information is more important than the global dependencies captured by transformer-based architectures, which include overall object structures.

Vision transformer can increase the performance of the Inception-only redshift prediction: Results of pcm-ViT-inception and plain-ViT-inception are remarkably better than the Inception-only redshift prediction. This demonstrates that for redshift prediction, in addition to the local information provided by the Inception modules, global dependency is necessary. Furthermore, the results show that pcm-ViT-inception achieves better outcomes compared to plain-ViT-inception.

Magnitude block can improve results: Results show that magnitude values corresponding to images can improve results significantly. These magnitudes are obtained by performing photometry on images. The improvement in results after adding the magnitude block demonstrates that both the Inception-only redshift prediction [34] and transformer-based models cannot gather these magnitudes from the images alone. For this reason, in our proposed fine-tuning architecture, the magnitude block is added to pcm-ViT-inception and plain-ViT-inception. The results demonstrate the capability of our fine-tuning architecture compared to other approaches.

Proposed stroM Es Outperform Henghes et al. [34]: Both proposed stroM E models outperform Henghes et al. [34]. The key difference between the proposed stroM E and Henghes et al. [34] is the use of a transformer-based model in fine-tuning. This demonstrates that, in addition to local information, capturing global dependencies and general patterns in the data is crucial for accurate redshift prediction.

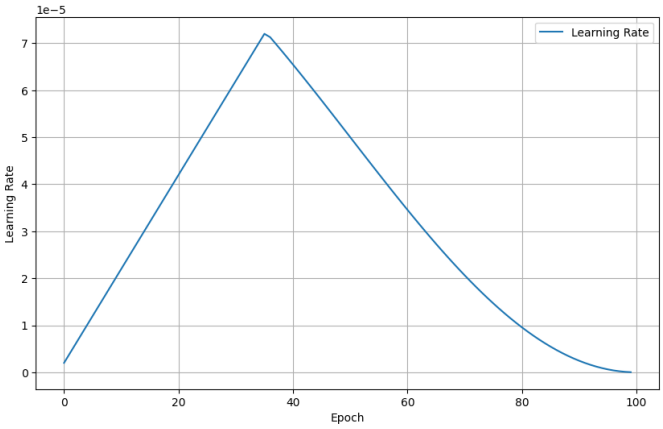


Fig. 8: Learning rate for training during the second experiment.

E. Second Experiment

In this experiment, we aim to evaluate the performance of pcm- stroM E and the baseline model [34], utilizing 100% of the data for both pretraining and fine-tuning. lthough the baseline model is a supervised learning approach and could potentially benefit from the increased amount of labeled data, our results indicate that pcm- stroM E consistently outperforms the baseline. Table V demonstrates the superiority of pcm- stroM E across evaluated metrics.

T BLE V: Comparison of Baseline Model and pcm- stroM E Performance using 100% of data.

Metric	Baseline Model	pcm- stroM E
MSE	0.00037	**0.00033**
M E	0.01302	**0.01267**
Bias	**-0.00157**	0.00191
Precision	0.01192	**0.01171**
R^2	0.96899	**0.97239**

1) Training Configuration: The configuration for pretraining remains consistent with the first experiment. For fine-tuning and training the baseline, the training scheduler adheres to the configuration illustrated in Figure 8. dditionally, Gaussian noise augmentation has been increased to 0.20 to further mitigate overfitting, leading to more stable training outcomes.

IV. Conclusions nd Future Works

In this paper, we employ a masked autoencoder—an efficient self-supervised learning method based on different transformer layers—for pretraining. To enhance the extraction of local information, we propose a novel hybrid fine-tuning method using inception modules for redshift prediction on the SDSS survey. Extensive experiments on various architectures constructed with vision transformers and CNNs demonstrate the lack of locality in transformer layers and the superiority of our method in addressing this issue. Based on the results, stroM E proves to be a successful redshift prediction method compared to the other methods tested in this paper.

In the next step, we aim to test our method on a broader range of downstream tasks to further demonstrate its capability and generality. We plan to extend the second experiment by evaluating all architectures mentioned in the first experiment using the full dataset. dditionally, we will conduct experiments to explore the effect of various mask ratios during pretraining on astrophysical image data. Furthermore, we intend to compare our methods with traditional approaches commonly used in astrophysics for redshift prediction.

V. cknowledgements

We thank the University of Virginia Computer Science Department, the Biocomplexity Institute, and the Department of Energy Grant DE-SC0023452: "F IR Surrogate Benchmarks Supporting I and Simulation Research" for partial support.

References

[1] Hubble, E. (1929). relation between distance and radial velocity among extra-galactic nebulae. Proceedings of the national academy of sciences, 15(3), 168-173.

[2] Dey, B., ndrews, B. H., Newman, J. ., Mao, Y. Y., Rau, M. M., & Zhou, R. (2022). Photometric redshifts from SDSS images with an interpretable deep capsule network. Monthly Notices of the Royal stronomical Society, 515(4), 5285-5305.

[3] Beck, R., Dobos, L., Budavári, T., Szalay, . S., & Csabai, I. (2016). Photometric redshifts for the SDSS Data Release 12. Monthly Notices of the Royal stronomical Society, 460(2), 1371-1381.

[4] Fan, X., McGreer, I., Patej, ., lvarez, ., Choi, Y., Jannuzi, B. T., ... & Zaritsky, D. (2019). Overview of the DESI Legacy Imaging Surveys.

[5] ihara, H., rimoto, N., rmstrong, R., rnouts, S., Bahcall, N. ., Bickerton, S., ... & Yuma, S. (2018). The Hyper Suprime-Cam SSP survey: overview and survey design. Publications of the stronomical Society of Japan, 70(SP1), S4.

[6] Salvato, M., Hasinger, G., Ilbert, O., Zamorani, G., Brusa, M., Scoville, N. Z., ... & Zamojski, M. (2008). PHOTOMETRIC REDSHIFT ND CL SSIFIC TION FOR THE XMM–COSMOS SOURCES. The strophysical Journal, 690(2), 1250.

[7] D'Isanto, ., & Polsterer, K. L. (2018). Photometric redshift estimation via deep learning-generalized and pre-classification-less, image based, fully probabilistic redshifts. stronomy & strophysics, 609, 111.

[8] Loshchilov, I., & Hutter, F. (2017). Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101.

[9] Pasquet, J., Bertin, E., Treyer, M., rnouts, S., & Fouchez, D. (2019). Photometric redshifts from SDSS images using a convolutional neural network. stronomy & strophysics, 621, 26.

[10] Rastegarnia, F., Mirtorabi, M. T., Moradi, R., Vafaei Sadr, ., & Wang, Y. (2022). Deep learning in searching the spectroscopic redshift of quasars. Monthly Notices of the Royal stronomical Society, 511(3), 4490-4499.

[11] Pasquet-Itam, J., & Pasquet, J. (2018). Deep learning approach for classifying, detecting and predicting photometric redshifts of quasars in the Sloan Digital Sky Survey stripe 82. stronomy & strophysics, 611, 97.

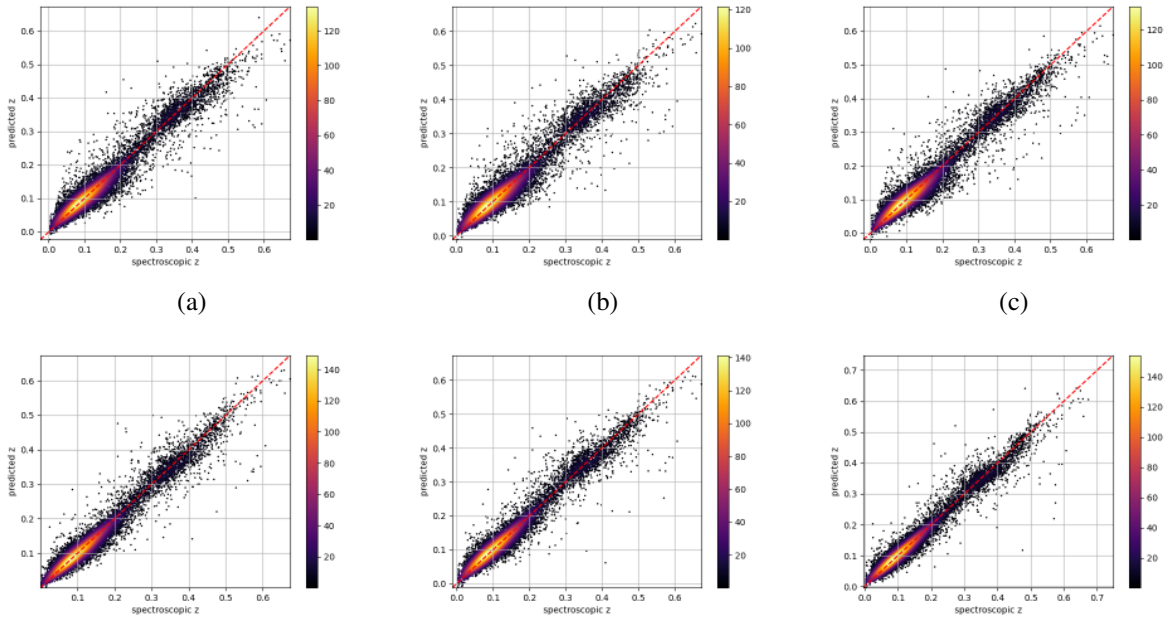
[12] Sandeep, V. Y., Sen, S., & Santosh, K. (2021, July). nalyzing and processing of astronomical images using deep learning techniques. In 2021 IEEE international conference on electronics, computing and communication technologies (CONECCT) (pp. 01-06). IEEE.

[13] Syarifudin, M. R. I., Hakim, M. I., & rifyanto, M. I. (2019, May). pplying deep neural networks (dnn) for measuring photometric redshifts from galaxy images: Preliminary study. In Journal of Physics: Conference Series (Vol. 1231, No. 1, p. 012013). IOP Publishing.

[14] Schuldt, S., Suyu, S. H., Cañameras, R., Taubenberger, S., Meinhardt, T., Leal-Taixé, L., & Hsieh, B. C. (2021). Photometric redshift estimation with a convolutional neural network: NetZ. stronomy & strophysics, 651, 55.

[15] Lin, J. Y. Y., Liao, S. M., Huang, H. J., Kuo, W. T., & Ou, O. H. M. (2021). Galaxy morphological classification with efficient vision transformer. arXiv preprint arXiv:2110.01024.

[16] Jing, L., & Tian, Y. (2020). Self-supervised visual feature learning with deep neural networks: survey. IEEE transactions on pattern analysis and machine intelligence, 43(11), 4037-4058.



(a) (b) (c)

(d) (e) (f)

Fig. 9: (a) plain-ViT, (b) from-scratch plain-ViT, (c) pcm-ViT (d) from-scratch pcm-ViT, (e) Inception-only redshift prediction, and (f) Henghes et al. [34] model.

[17] Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., & Efros, . . (2016). Context encoders: Feature learning by inpainting. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2536-2544).

[18] Hayat, M. ., Stein, G., Harrington, P., Lukić, Z., & Mustafa, M. (2021). Self-supervised representation learning for astronomical images. The strophysical Journal Letters, 911(2), L33.

[19] Lanusse, F., Parker, L., Golkar, S., Cranmer, M., Bietti, ., Eickenberg, M., ... & Ho, S. (2023). stroCLIP: Cross-Modal Pre-Training for stronomical Foundation Models. arXiv preprint arXiv:2310.03024.

[20] Shen, G., Zou, Z., Luo, . L., Hong, S., & Kong, X. (2023). Galaxy Morphology Classification Model Based on Momentum Contrastive Learning. Publications of the stronomical Society of the Pacific, 135(1052), 104501.

[21] Stein, G., Harrington, P., Blaum, J., Medan, T., & Lukic, Z. (2021). Self-supervised similarity search for large scientific datasets. arXiv preprint arXiv:2110.13151.

[22] Donoso-Oliva, C., Becker, I., Protopapas, P., Cabrera-Vives, G., Vishnu, M., & Vardhan, H. (2023). STROMER- transformer-based embedding for the representation of light curves. stronomy & strophysics, 670, 54.

[23] Schroff, F., Kalenichenko, D., & Philbin, J. (2015). Facenet: unified embedding for face recognition and clustering. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 815-823).

[24] Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020, November). simple framework for contrastive learning of visual representations. In International conference on machine learning (pp. 1597-1607). PMLR.

[25] Chen, X., & He, K. (2021). Exploring simple siamese representation learning. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 15750-15758).

[26] He, K., Chen, X., Xie, S., Li, Y., Dollár, P., & Girshick, R. (2022). Masked autoencoders are scalable vision learners. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 16000-16009).

[27] Dosovitskiy, ., Beyer, L., Kolesnikov, ., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N. (2020). n image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929.

[28] Ramachandran, P., Zoph, B., & Le, Q. V. (2017). Searching for activation functions. arXiv preprint arXiv:1710.05941.

[29] Hendrycks, D., & Gimpel, K. (2016). Gaussian error linear units (gelus). arXiv preprint arXiv:1606.08415.

[30] Xu, Y., Zhang, Q., Zhang, J., & Tao, D. (2021). Vitae: Vision transformer advanced by exploring intrinsic inductive bias. dvances in neural information processing systems, 34, 28522-28535.

[31] Wang, D., Zhang, Q., Xu, Y., Zhang, J., Du, B., Tao, D., & Zhang, L. (2022). dvancing plain vision transformer toward remote sensing foundation model. IEEE Transactions on Geoscience and Remote Sensing, 61, 1-15.

[32] Sun, X., Wang, P., Lu, W., Zhu, Z., Lu, X., He, Q., ... & Fu, K. (2022). RingMo: remote sensing foundation model with masked image modeling. IEEE Transactions on Geoscience and Remote Sensing.

[33] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., nguelov, D., ... & Rabinovich, . (2015). Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1-9).

[34] Henghes, B., Thiyagalingam, J., Pettitt, C., Hey, T., & Lahav, O. (2022). Deep learning methods for obtaining photometric redshift estimations from images. Monthly Notices of the Royal stronomical Society, 512(2), 1696-1709.

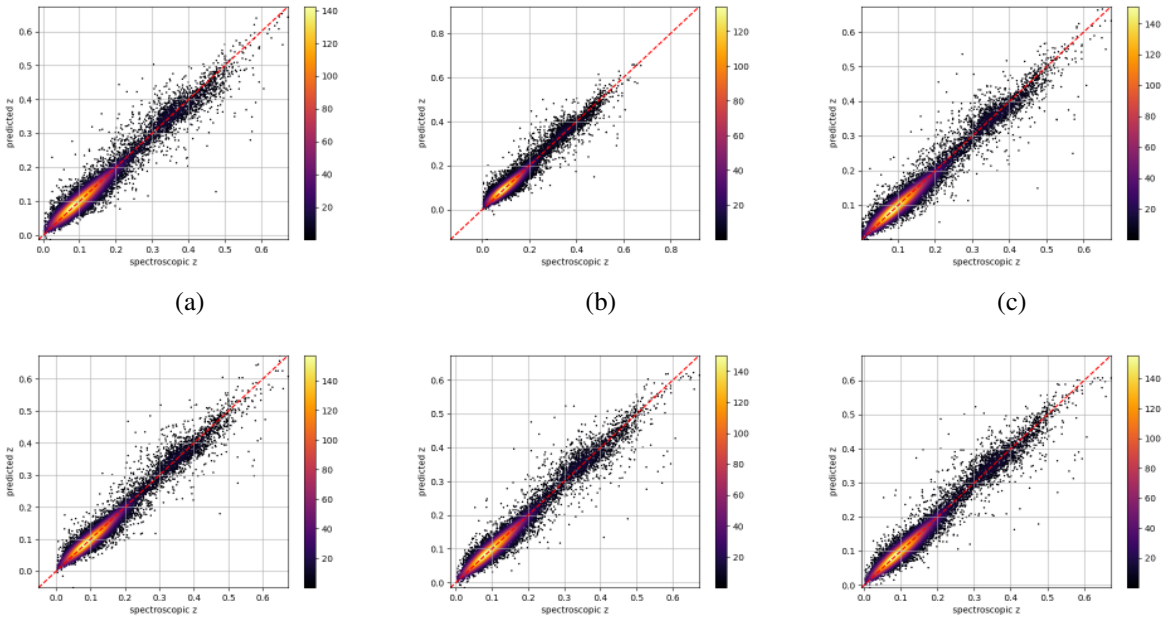
[35] Gunn, J. E., Carr, M., Rockosi, C., Sekiguchi, M., Berry, K., Elms, B., ... & Brinkman, J. (1998). The sloan digital sky survey photometric camera. The stronomical Journal, 116(6), 3040.

[36] Gunn, J. E., Siegmund, W. ., Mannery, E. J., Owen, R. E., Hull, C. L., Leger, R. F., ... & Wang, S. I. (2006). The 2.5 m telescope of the sloan digital sky survey. The stronomical Journal, 131(4), 2332.

[37] York, D. G., delman, J., nderson Jr, J. E., nderson, S. F., nnis, J., Bahcall, N. ., ... & Yasuda, N. (2000). The sloan digital sky survey: Technical summary. The stronomical Journal, 120(3), 1579.

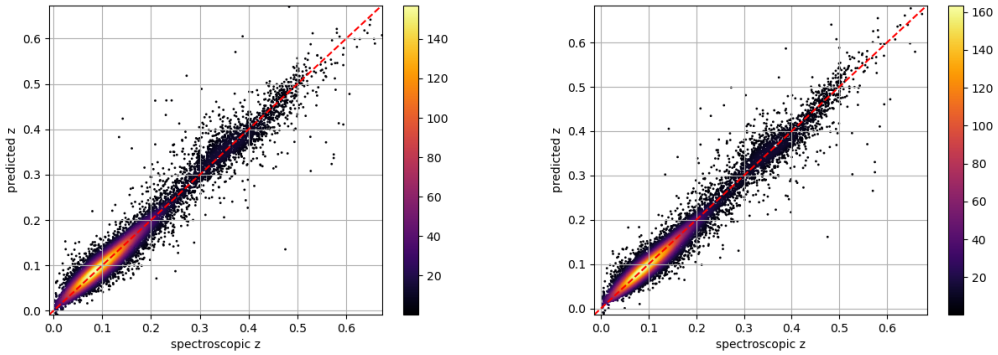
[38] Ginsburg, ., Sipőcz, B. M., Brasseur, C. E., Cowperthwaite, P. S., Craig, M. W., Deil, C., ... & Woillez, J. (2019). stroquery: an astronomical web-querying package in Python. The stronomical Journal, 157(3), 98.

[39] He, T., Zhang, Z., Zhang, H., Zhang, Z., Xie, J., & Li, M. (2019). Bag of tricks for image classification with convolutional neural networks. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 558-567).



(a) (b) (c) (d) (e) (f)

Fig. 10: (a) plain-ViT-magnitude, (b) from-scratch plain-ViT-magnitude, (c) pcm-ViT-magnitude, (d) from-scratch pcm-ViT-magnitude, (e) plain-ViT-inception, and (f) pcm-ViT-inception.



(a) (b)

Fig. 11: (a) Proposed plain- stroM E, and (b) Proposed pcm- stroM E.

[40] Xie, Z., Zhang, Z., Cao, Y., Lin, Y., Wei, Y., Dai, Q., & Hu, H. (2023). On data scaling in masked image modeling. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 10365-10374).

[41] Ba, J. L., Kiros, J. R., & Hinton, G. E. (2016). Layer normalization. arXiv preprint arXiv:1607.06450.

[42] Cohen, J. G., Hogg, D. W., Blandford, R., Cowie, L. L., Hu, E., Songaila, ., ... & Richberg, K. (2000). Caltech faint galaxy redshift survey. x. a redshift survey in the region of the hubble deep field north. The strophysical Journal, 538(1), 29.

[43] Ilbert, O., rnouts, S., Mccracken, H. J., Bolzonella, M., Bertin, E., Le Fèvre, O., ... & Vergani, D. (2006). ccurate photometric redshifts for the CFHT legacy survey calibrated using the VIMOS VLT deep survey. stronomy & strophysics, 457(3), 841-856.

[44] Si, C., Yu, W., Zhou, P., Zhou, Y., Wang, X., & Yan, S. (2022). Inception transformer. dvances in Neural Information Processing Systems, 35, 23495-23509.