Inferring Iterated Function Systems Approximately from Fractal Images

Haotian Liu[1,2*], **Dixin Luo**[2,3], **Hongteng Xu**[1,4†]
[1]Gaoling School of Artificial Intelligence, Renmin University of China, Beijing
[2]School of Computer Science and Technology, Beijing Institute of Technology, Beijing
[3]Key Laboratory of Artificial Intelligence, Ministry of Education, Shanghai
[4]Beijing Key Laboratory of Big Data Management and Analysis Methods, Beijing
hongtengxu@ruc.edu.cn

Abstract

As an important mathematical concept, fractals commonly appear in nature and inspire the design of many artistic works. Although we can generate various fractal images easily based on different iterated function systems (IFSs), inferring an IFS from a given fractal image is still a challenging inverse problem for both scientific research and artistic design. In this study, we explore the potential of deep learning techniques for this problem, learning a multi-head auto-encoding model to infer typical IFSs (including Julia set and L-system) from fractal images. In principle, the proposed model encodes fractal images in a latent space and decodes their corresponding IFSs based on the latent representations. For the fractal images generated by heterogeneous IFSs, we let them share the same encoder and apply two decoders to infer the sequential and non-sequential parameters of their IFSs, respectively. By introducing one more decoder to reconstruct fractal images, we can leverage large-scale unlabeled fractal images to learn the model in a semi-supervised way, which suppresses the risk of over-fitting. Comprehensive experiments demonstrate that our method provides a promising solution to infer IFSs approximately from fractal images. Code and supplementary file are available at https://github.com/HaotianLiu123/Inferring-IFSs-From-Fractal-Images.

1 Introduction

The study of fractal geometry can be traced back to the late 1960s when Benoit B. Mandelbrot systematically explored complex structures with self-similarity and formally introduced the term "fractal" [Mandelbrot and Mandelbrot, 1982]. As fascinating and important mathematical concepts, fractals are widely exhibited in the natural world and serve as inspiration for numerous artworks. In particular, with the continuous advancement of computer technology, the digital creation methods of fractal art offer new avenues for artistic innovation. Artists can utilize different iterated function systems (IFSs) to create intricate and exquisite fractals [Lindenmayer, 1968; Barnsley, 1988], and the fractals' distinctive geometric shapes and self-similarity features provide artists with endless inspiration, bestowing upon the uniqueness of their creations.

Besides creating fractals, the inverse problem of inferring IFSs from given fractal images holds significance in art design and even impacts scientific discovery. In particular, by inferring IFSs from the fractals, artists can explore self-similarity patterns hidden in natural scenes and create new fractal-based artworks accordingly [Oppenheimer, 1986]. Additionally, the inferred IFSs help render natural landscapes and lead to fractal-based image compression techniques [Pentland, 1984]. Moreover, many natural phenomena, e.g., the growth of filamentous organisms [Barry *et al.*, 2009; Lee, 2022] and the generation of crystal materials [Tsai and Mecholsky, 1991; Zhao *et al.*, 2018], can be described as fractals. Inferring IFSs from their images is important for building corresponding dynamic systems.

However, due to the uncertainty and complexity of fractal images and the associated IFSs, the IFS of a fractal image has a huge search space with an unknown intrinsic structure. As a result, it is always challenging to infer IFSs from fractal images in practice without sufficient prior knowledge. Moreover, there are many various IFSs generating fractals, e.g., Julia set of complex functions [Julia, 1918], L-systems [Lindenmayer, 1968], and so on. The heterogeneity of such IFSs further increases the difficulty of their inference task. Currently, some attempts have been made to infer IFSs automatically. The methods in [Jacquin, 1992; Hoskins and Vagners, 1992; Kapoor *et al.*, 2004] infer IFSs through image compression, and the generic methods in [Angeline, 1994] infer IFSs via heuristic searching. Focusing on L-systems, approximate inference can be achieved by search space shrinkage [Jürgensen and Lindenmayer, 1987] and sequential rule exploration [de la Higuera, 2005]. However, these methods are designed for specific IFSs and exhibit high computational complexity, whose performance is unsatisfactory when inferring heterogeneous IFSs.

To overcome the above challenges, we explore the potential of deep learning techniques for the inference problem of IFS, learning a multi-head autoencoder in a semi-supervised way to infer various IFSs approximately from fractal images.

*This work is mainly done when Haotian Liu works as a research intern at Gaoling School of Artificial Intelligence.

†Corresponding author.

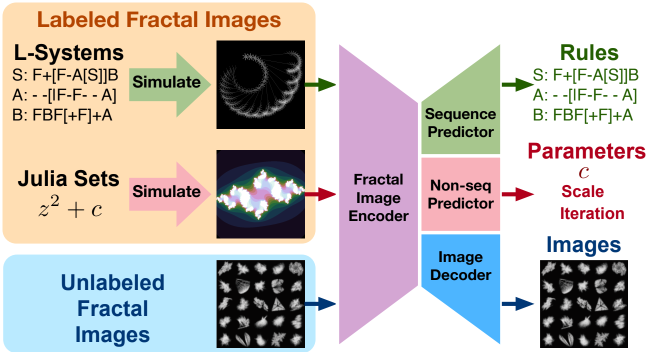


Figure 1: An illustration of our method.

As illustrated in Figure 1, our model encodes fractal images in a latent space and decodes the latent representations to infer the parameters of the corresponding IFSs. The fractal images can be generated by different IFSs, including Julia sets and L-systems. Because these IFSs have sequential and non-sequential parameters, our model applies a multi-head auto-encoding architecture, encoding the fractal images by the same encoder and inferring the sequential and non-sequential parameters by two different decoding heads, respectively. Furthermore, to suppress the risk of over-fitting, we further introduce an image decoding head to reconstruct the fractal images from their latent representations. As a result, besides using the fractal images simulated by known IFSs, we can leverage unlabeled fractal images (e.g., those in FractalDB [Kataoka *et al.*, 2022]) when training the model, which leads to a semi-supervised learning paradigm. We conduct comprehensive experiments, learning different models by different learning paradigms and comparing their quantitative and qualitative performance. Experimental results demonstrate that our method provides a promising solution to infer IFSs approximately from fractal images.

2 Related Work

2.1 Iterated Function System Inference

In general, fractals are geometrical objects that have self-similar and detailed structures at arbitrarily small scales, whose fractal dimension strictly exceeds their topological dimensions [Mandelbrot and Aizenman, 1979]. The fractals are often generated by different iterated function systems, like recursive compositions of complex functions (e.g., Julia sets [Barnsley, 1988]), chaos games [Barnsley and Vince, 2011], and recursive parallel rewriting systems based on specific formal grammars (i.e., L-systems [Lindenmayer, 1968]).

Some methods have been proposed to infer specific IFSs from fractal images. In particular, given a fractal image, the work in [Jacquin, 1992] first presents a partial IFS inference method to infer restricted IFSs for the local patches of the image, which leads to the well-known fractal-based image compression techniques. Following this strategy, some variants [Hoskins and Vagners, 1992; Kapoor *et al.*, 2004; Menassel *et al.*, 2020] are proposed to improve the efficiency of the method. Among them, the method in [Rinaldo and Zakhor, 1994] applies Wavelet transform to realize the extraction of IFS parameters, connecting IFS inference to multi-scale analysis. However, these methods often have high computational complexity and can only infer local and limited IFSs based on specialized knowledge in related fields. Focusing on L-systems, the methods in [Herman and Walker, 1972; de la Higuera, 2005; Bernard and McQuillan, 2023] try to estimate the grammar of L-systems based on a finite set of strings, which fail to infer L-systems based on fractal images. The genetic algorithm in [Angeline, 1994] applies a heuristic strategy to achieve image-based L-system inference, whose convergence and performance have no theoretical guarantees. Moreover, the above methods are designed for a specific kind of IFS. They do not provide a unified framework to infer heterogeneous IFSs and thus suffer from poor generalizability.

2.2 Neural Network-based Image Understanding

In the past ten years, deep learning has proven to be a powerful tool for image understanding, which extracts informative image representations via learning neural networks. Typically, convolutional neural networks (CNNs), like AlexNet [Krizhevsky *et al.*, 2012], VggNet [Simonyan and Zisserman, 2015], ResNet [He *et al.*, 2016], and their variants [Howard *et al.*, 2017; Huang *et al.*, 2017], achieve encouraging performance in large-scale image classification tasks, which provide valuable backbones for image representation and understanding. Recently, Transformer [Vaswani *et al.*, 2017] has also been applied to vision tasks, leading to the ViT model [Dosovitskiy *et al.*, 2020]. These models represent images semantically in latent spaces, and the latent representations can be used to support various downstream tasks, e.g., conditional image generation [Van den Oord *et al.*, 2016; Kim *et al.*, 2022; Li *et al.*, 2023] and cross-modal generation [Li *et al.*, 2021; Li *et al.*, 2022]. For example, connecting these models with sequential models [Radford *et al.*, 2018; Raffel *et al.*, 2020] leads to an auto-encoding architecture for conditional text generation [Ramesh *et al.*, 2022; Mai *et al.*, 2020]. However, whether these models can encode fractal images well or not and how to train the models to provide sufficient information to infer IFSs are open problems not investigated yet, which motivates this study.

3 Proposed Method

3.1 Multi-head Auto-encoding Architecture

As mentioned before, fractal images can be generated by heterogeneous IFSs. Take the Julia set [Julia, 1918] for quadratic complex polynomials as an example. We can generate the Julia set, a point set with a self-similarity structure defined on the complex plane, by recursively applying the following complex function:

$$z_{n+1} = z_n^2 + c, \quad n = 0, 1, 2, \cdots, \tag{1}$$

where the offset $c \in \mathbb{C}$ is a complex number. z_0 is the initial input, which is fixed as $0+0j$ in this study. Given c and an initial z_0, a series of complex numbers can be generated, and the Julia set is constructed by the points remaining bounded during the recursive process. Applying the algorithm provided in [Hussein *et al.*, 1999], we can generate a Julia set by applying (1) N times given specific c and z_0. In summary, each

Symbol	Definition
F	Draw forwards
[,]	Push / pop state
+, −	Rotate by +/- angle
!	Negate angle
|	Increment angle by 180°

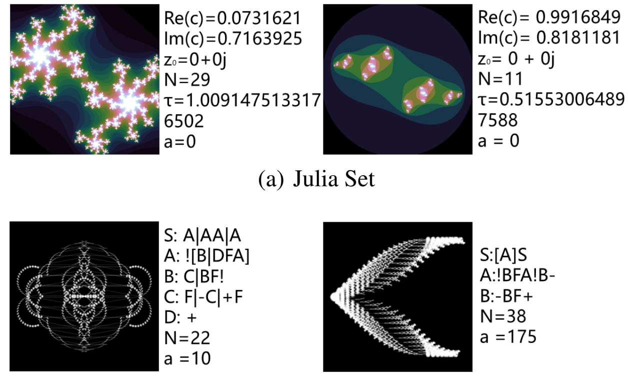
Table 1: The definitions of operation symbols in L-system.

fractal image of Julia set, denoted as $\boldsymbol{I}^{(J)}$, is associated with two parameters $\boldsymbol{y} = \{\mathrm{Re}(c), \mathrm{Im}(c)\}$, where $c \in \mathbb{C}$, $\mathrm{Re}(c)$ and $\mathrm{Im}(c)$ are its real and imaginary parts.

While the fractal images of the Julia set are determined by non-sequential parameters, the fractal images generated by L-systems are associated with sequential parameters. In particular, an L-system [Lindenmayer, 1968] is a parallel rewriting system and a type of formal grammar, which can plot fractal images by recursively implementing a set of rules. Mathematically, it can be defined as a tuple $G = (\mathcal{V}, \omega, \mathcal{R})$. $\mathcal{V}$ is an alphabet set, which encompasses symbols containing both elements that can be replaced (variables) and those that cannot be replaced (constants or terminals). Each symbol in $\mathcal{V}$ is associated with a rule, so the cardinality of $\mathcal{V}$ determines the number of rules. ω is the initiator, which is a string of symbols from $\mathcal{V}$ defining the initial system state. $\mathcal{R} = \{r_i\}_{i=1}^{|\mathcal{V}|}$ is a set of rules determining how to replace the variables with combinations of constants and other variables. Each rule r is a sequence formulated as "$p : s$". $p \in \mathcal{V}$ is called predecessor, which corresponds to the variable to be replaced in the next iteration. $s \subset \mathcal{V} \cup \mathcal{A}$ is a string determines the successor used to replace p. Here, $\mathcal{A} = \{F, [,], +, -, !, |\}$ is a set of operations applying to the variables in $\mathcal{V}$, whose definitions are in Table 1. Accordingly, s means applying a series of operations to some specific variables and replacing p with the operations' result. Obviously, for each fractal image generated by an L-system, denoted as $\boldsymbol{I}^{(L)}$, the corresponding rule set $\mathcal{R} = \{r_i\}_{i=1}^{|\mathcal{V}|}$ is its parameters. By concatenating the rules, we can formulate the parameters as a symbol sequence, denoted as $\boldsymbol{s}$, whose vocabulary set is $\mathcal{V} \cup \mathcal{A}$.

Figure 2 shows typical fractal images of Julia set and L-system, respectively. According to the above analysis, we can find that the parameters of Julia set are non-sequential, while those of L-system are sequential. Additionally, we can implement the fractal images with different iteration numbers and further apply geometrical transformations (e.g., zoom-in, zoom-out, and rotation) to increase their diversity. Both of the IFSs can take the number of iterations $N \in \mathbb{N}$, the scaling coefficient $\tau \in (0, \infty)$, and the rotation angle $a \in [0°, 360°)$ as additional model parameters. Therefore, we need to build a model to predict the sequential and non-sequential parameters jointly.

To achieve this aim, we design a model with multi-head auto-encoding architecture. Specifically, the model consists of one image encoder and three decoders. The encoder, denoted as $f : \mathcal{I} \mapsto \mathcal{Z}$, maps fractal images to a d-dimensional latent space $\mathcal{Z} \subset \mathbb{R}^d$, i.e., $\boldsymbol{z} = f(\boldsymbol{I})$. Based on the latent representations of fractal images, the first decoder g_1 predicts the sequential rules of the L-system, incorporating both the number of iterations and rotation angle parameters into the rules for joint prediction. The second decoder g_2 predicts the non-sequential parameters of the Julia set. Additionally, the third decoder $g_3 : \mathcal{Z} \mapsto \mathcal{I}$ aims to reconstruct each input image based on its latent representation. Typically, we can implement $\{f, g_2, g_3\}$ by classic CNNs or Transformer encoders and implement g_1 by a sequential model like recurrent neural networks (RNNs) or Transformer decoders.



(a) Julia Set

(b) L-system

Figure 2: Typical fractal images of Julia set and L-system. For each image, the corresponding parameters are provided.

The first two decoders work for inferring IFSs from latent representations. Connecting f with g_1 and g_2 leads to the target model in the testing phase. To suppress the risk of over-fitting, we apply two mechanisms. Firstly, we let the fractal images generated by different IFSs share the same encoder, mapping them to the same latent space. Secondly, the third decoder works to construct a regularizer — by penalizing the reconstruction loss of input fractal images, we can ensure that the latent representations preserve sufficient semantic information for the images. Note that the third decoder and the associated reconstruction loss do not rely on the parameters of IFSs, so they are applicable for unlabeled fractal images. As a result, the utilization of large-scale unlabeled fractal images helps improve the generalization power of the target model, which leads to the following semi-supervised learning paradigm.

3.2 Semi-supervised Learning Paradigm

Denote the labeled fractal images of Julia set and L-system as two sets, i.e., $\mathcal{D}_J = \{\boldsymbol{I}^{(J)}, \boldsymbol{y}^{(J)}\}$ and $\mathcal{D}_L = \{\boldsymbol{I}^{(L)}, \boldsymbol{s}^{(L)}\}$, where $\boldsymbol{y}^{(J)}$ and are non-sequential parameters of the Julia Set and $\boldsymbol{s}^{(L)}$ represents the sequential parameters of L-system. Additionally, we denote the unlabeled fractal images we collected as $\mathcal{D}_U = \{\boldsymbol{I}^{(U)}\}$. Given such training data, we can learn our multi-head autoencoder by considering the following three losses.

Sequential Parameter Prediction

For the sequential parameters, we predict the element in each sequence in an autoregressive manner, leading to the following cross-entropy loss:

$$L_1(f, g_1) := \sum\nolimits_{(\boldsymbol{I},\boldsymbol{s}) \in \mathcal{D}_L} \sum\nolimits_{s_i \in \boldsymbol{s}} \mathrm{CE}(s_i, g_1(f(\boldsymbol{I}), \boldsymbol{s}_i)), \quad (2)$$

where $\boldsymbol{s}_i = \{s_j\}_{j=1}^{i-1}$ is the historical elements before s_i.

Non-sequential Parameter Prediction

For the non-sequential parameters, we take the mean-square-error (MSE) of their estimation as the loss function, i.e.,

$$L_2(f, g_2) := \sum\nolimits_{(\boldsymbol{I},\boldsymbol{y})\in\mathcal{D}_J} \|g_2(f(\boldsymbol{I})) - \boldsymbol{y}\|_2^2. \tag{3}$$

Fractal Image Reconstruction

Finally, we consider the MSE of the reconstructed fractal images during training, i.e.,

$$L_3(f, g_3) := \sum\nolimits_{\boldsymbol{I}\in\mathcal{D}_J\cup\mathcal{D}_L\cup\mathcal{D}_U} \|g_3(f(\boldsymbol{I})) - \boldsymbol{I}\|_F^2, \tag{4}$$

where $\|\cdot\|_F$ is the Frobenius norm of matrix.

Considering the above three loss functions jointly leads to the proposed learning problem, i.e.,

$$\min_{f,g_1,g_2,g_3} \alpha L_1(f, g_1) + (1-\alpha)L_2(f, g_2) + \beta L_3(f, g_3), \tag{5}$$

where $\alpha \in [0, 1]$ achieves a trade-off between the sequential and non-sequential prediction, and $\beta > 0$ controls the significance of the regularization. We solve (5) efficiently by stochastic gradient descent [Robbins and Monro, 1951].

4 Experiment

4.1 Implementation Details

Data Preparation

To demonstrate the feasibility of our inference method and evaluate its performance, we construct a fractal image dataset, which consists of the following three subsets.

- **Labeled fractal images of Julia set.** For Julia set, we randomly generate 10,000 images. All of the parameters are sampled randomly and uniformly from a specific range, i.e., both the real and imaginary parts of c are in the range $[-1, 1]$, $N \in [10, 30]$, the scaling coefficient $\tau \in [0.5, 1.5]$.
- **Labeled fractal images of L-system.** For L-system, we generate 10,065 fractal images based on 59 different rules that are proven to generate reasonable fractals. The rules are sequences generated with the same vocabulary set $\mathcal{V} = \{S, A, B, C, D, E\}$. We generate 59 fixed categories of grammar. To make the dataset more diverse, we set $a \in [0, 175°]$ and choose 5 different iteration numbers for each rule.
- **Unlabeled fractal images.** We apply the FractalDB60 dataset proposed in [Kataoka *et al.*, 2022]. It contains 60 categories of fractal images, each with 1000 instances generated based on the corresponding unknown IFSs. To match the quantity of the data we generated, we randomly selected 200 images from each category, resulting in a total of 12,000 images.

For the labeled fractal images, we split them into training and testing sets. All the unlabeled fractal images are used for training. The algorithms generating the labeled fractal images are shown in the supplementary file.

Model Architectures

For the proposed multi-head autoencoder, we consider different model architectures and analyze their impacts on learning results. Specifically, we implement the encoder f based on four representative architectures, including VGG16 [Simonyan and Zisserman, 2015], Resnet50 [He *et al.*, 2016], DenseNet [Huang *et al.*, 2017], and ViT [Dosovitskiy *et al.*, 2020]. For the three decoders, we implement g_1 based on the recurrent neural network (RNN) in [Xu *et al.*, 2015], implement g_2 based on the convolutional neural network (CNN) within the encoder, and implement g_3 based on the image generator in [Goodfellow *et al.*, 2014].

Hyperparameter Settings

We implement our method by PyTorch and conduct all experiments on a single NVIDIA 3090 GPU. We train our models by Adam [Kingma and Ba, 2014], and we set the batch size to be 16, the epochs to be 150. The learning rate is set to be 10^{-4} for $\{f, g_2, g_3\}$ and 4×10^{-4} for g_1, respectively. The clip gradient is set to be 5. As for the hyperparameters in (5), we set $\beta = 0.1$ and $\alpha = 0.95$ for the first 120 epochs, and then solely update the model for predicting non-sequential parameters in subsequent epochs.

Evaluation Metrics

For the non-sequential parameters, we utilize the Mean Absolute Error (MAE) to evaluate their estimation results. For the sequential rules of L-system, we evaluate the quality of machine-generated rules based on the commonly-used BLEU [Papineni *et al.*, 2002] and ROUGE (R@1 and R@L) [Lin, 2004]. Additionally, based on the inferred IFSs, we can simulate fractal images and compare them with those generated by the ground truth IFSs. The objective image quality assessment metrics, like SSIM (Structural Similarity Index) [Wang *et al.*, 2004] and LPIPS (Learned Perceptual Image Patch Similarity) [Zhang *et al.*, 2018], can be applied. Specifically, SSIM measures the similarity based on the luminance, contrast, and structure factors, while LPIPS measures the human perceptual similarity between images.

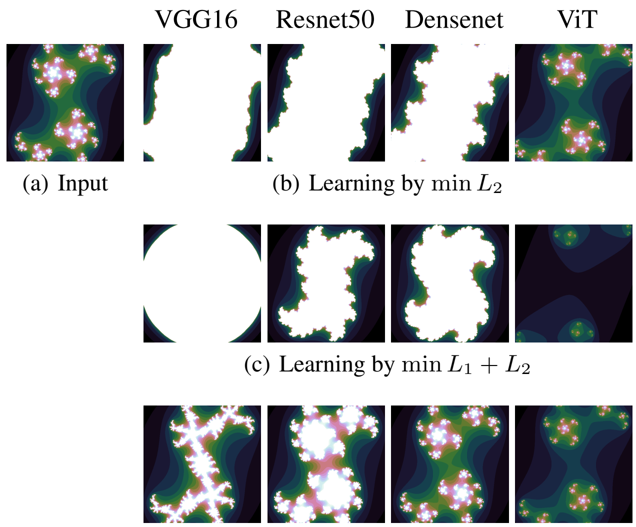
4.2 Quantitative and Qualitative Results

Comparisons for Learning Paradigms

Our quantitative results are shown in Table 2. We can find that when separately training different IFSs, the model of one IFS is not applicable for inferring other IFSs in general. For example, we can't apply the model trained for Julia set to infer L-system, as the model parameters are not shared. Additionally, the model trained for a single IFS suffers from a high risk of over-fitting due to the insufficiency of training data. However, when training an inference model for the two IFSs jointly (i.e., minimizing the L_1 in (2) and the L_2 in (3) jointly), the model is not consistently better than those trained separately on the proposed evaluation measurements because the heterogeneity of the training data leads to a much more difficult learning task. Our semi-supervised learning paradigm overcomes the drawbacks of the above two paradigms, which achieves the best performance in most situations. In particular, by introducing the reconstruction task for both labeled and unlabeled fractal images (i.e., the L_3

Encoder	L_1	L_2	L_3	Julia Set						L-System						
				MAE↓				Image Similarity		MAE↓		Sequence Quality↑			Image Similarity	
				Re(c)	Im(c)	N	τ	SSIM↑	LPIPS↓	a	N	BLEU	R@1	R@L	SSIM↑	LPIPS↓
VGG16	×	✓	×	0.1840	0.2170	5.4311	0.1533	0.5045	0.4996	—	—	—	—	—	—	—
	✓	×	×	—	—	—	—	—	—	5.972	2.865	89.32	91.10	93.42	0.9122	0.1394
	✓	✓	×	0.2450	0.2274	1.7099	0.1964	0.5706	0.4486	5.085	2.610	88.90	93.22	92.43	0.9095	0.1481
	✓	✓	✓	**0.0770**	**0.1153**	**1.2314**	**0.0849**	**0.7012**	**0.2948**	**3.670**	**2.454**	**94.84**	**95.52**	**95.37**	**0.9178**	**0.1312**
Resnet50	×	✓	×	0.1610	0.1547	1.8569	0.1348	0.6358	0.3756	—	—	—	—	—	—	—
	✓	×	×	—	—	—	—	—	—	4.165	2.593	90.55	**94.26**	**93.87**	0.9108	0.1404
	✓	✓	×	0.1219	0.1414	1.4730	0.2816	0.5412	0.4127	9.700	2.770	87.89	92.96	92.52	0.9030	0.1530
	✓	✓	✓	**0.0984**	**0.0822**	**1.4057**	**0.0567**	**0.7237**	**0.2733**	**4.133**	**2.497**	**91.40**	93.34	93.33	**0.9123**	**0.1393**
DenseNet	×	✓	×	0.1223	0.2072	1.9311	0.0827	0.6430	0.3743	—	—	—	—	—	—	—
	✓	×	×	—	—	—	—	—	—	3.420	**2.260**	91.73	**94.54**	94.25	0.9154	0.1282
	✓	✓	×	0.0973	0.0853	1.5406	0.0531	0.7269	0.2728	4.194	2.505	90.21	94.29	93.92	0.9171	0.1364
	✓	✓	✓	**0.0497**	**0.0543**	**1.3220**	**0.0348**	**0.7761**	**0.1905**	**3.290**	2.426	**93.34**	94.29	**94.28**	**0.9179**	**0.1139**
ViT	×	✓	×	**0.0915**	0.2285	1.4751	0.0645	0.6638	0.3479	—	—	—	—	—	—	—
	✓	×	×	—	—	—	—	—	—	33.32	8.515	65.50	**88.62**	**85.34**	0.8997	0.2575
	✓	✓	×	0.1962	**0.1966**	**1.4361**	**0.0641**	0.6557	0.3610	34.42	8.575	65.03	88.48	85.21	0.8821	0.2407
	✓	✓	✓	0.0991	0.2108	1.5664	0.0653	**0.6690**	**0.3129**	**32.58**	**8.505**	**65.86**	88.16	85.19	**0.9026**	**0.2394**

Table 2: Quantitative experimental results for various model architectures and learning paradigms.

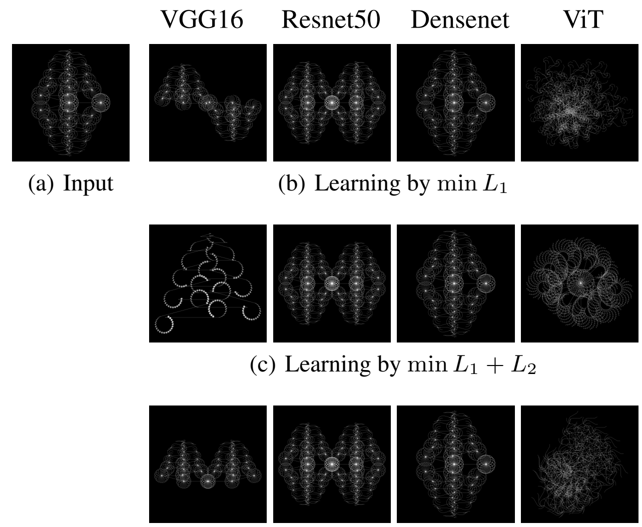


(a) Input (b) Learning by $\min L_2$

(c) Learning by $\min L_1 + L_2$

(d) Learning by $\min L_1 + L_2 + L_3$

Figure 3: Visual comparisons for different models and learning paradigms given a Julia Set.



(a) Input (b) Learning by $\min L_1$

(c) Learning by $\min L_1 + L_2$

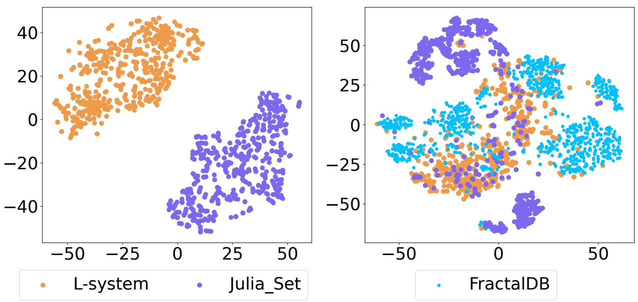
(d) Learning by $\min L_1 + L_2 + L_3$

Figure 4: Visual comparisons for different models and learning paradigms given an L-system fractal image.

in (4)), our method mitigates the insufficiency of data and thus suppresses the risk of over-fitting. The reconstruction task penalizes the loss of information, ensuring the latent representations of fractal images are semantically meaningful.

Besides the numerical results, we apply the inferred IFSs to simulate fractal images and compare the generated images with the input ones. Figures 3 and 4 show the fractal images generated by different models under different paradigms. These results further demonstrate the superiority of the proposed semi-supervised learning paradigm. When learning the models separately or without the reconstruction loss, they often infer the IFSs with low precision. As a result, their generated fractal images are significantly different from the input ones, and some models even fail to generate fractals. In contrast, applying the semi-supervised learning paradigm improves the model performance consistently across different model architectures, which helps the models infer IFSs with higher precision and thus generate fractal images more similar to the input images.

The advantage of the proposed semi-supervised learning paradigm can also be verified by the distribution of latent representations. In particular, given the latent representations of fractal images, we employed t-SNE [Maaten and Hinton, 2008] to visualize them in 2D space. As shown in Figure 5, without the help of unlabeled fractal images and the reconstruction-based regularization, the t-SNE plot of Julia set is separated from that of L-system. In other words, the latent distribution of the fractal images has two separate modal-



(a) $\min L_1 + L_2$ (b) $\min L_1 + L_2 + L_3$

Figure 5: The t-SNE plots of the latent representations of fractal images. In this experiment, we apply the model with Densenet encoder and randomly select 500 Julia Set images, 500 L-system fractal images, and 1,000 FractalDB images to compute the t-SNE plot.

ities and the latent representations between the two modalities are unavailable. It implies that the model does not overcome the heterogeneity of these two IFSs and suffers from a high risk of over-fitting. In contrast, when applying the unlabeled fractal images to train our model, $i)$ the latent representations of the unlabeled fractal images fill the blank spaces between the modalities of the Julia set and L-system, $ii)$ some latent representations of Julia set are mixed with those of L-system, and $iii)$ the clustering structure within each dataset becomes significant. For our model, these two phenomena imply an improvement in generalizability.

Impact of Encoder Architecture

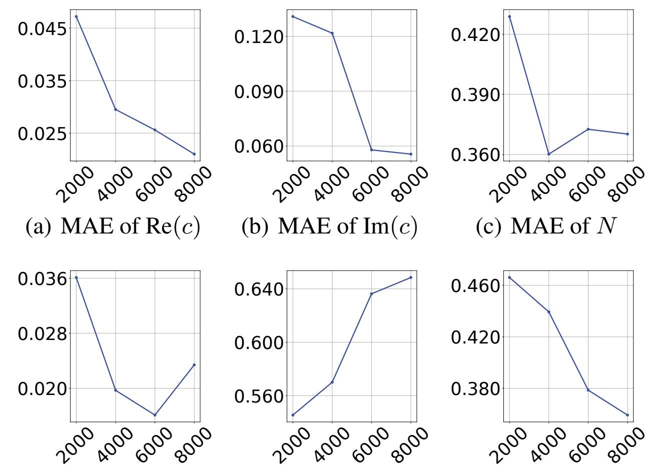
According to the above results, we can further analyze the impacts of different model architectures. We can find that among the four model architectures, Densenet achieves the best performance in most situations, as shown in Table 2. In particular, the learned Densenet can achieve encouraging numerical results when inferring IFSs, and the inferred IFSs can generate reasonable fractal images that are similar to those generated by the ground truth IFSs, as shown in Figures 3 and 4. Therefore, we apply Densenet as the default architecture of encoder. It should be noted that the commonly-used ViT architecture does not perform well when inferring IFSs. In our opinion, a potential reason for this phenomenon is that the fractal structure is a global spatial pattern of each fractal image. The convolution neural networks (i.e., VGG16, Resnet50, and Densenet) do not change the spatial relations among image pixels, while ViT tokenizes each fractal image as a sequence of local patches. The local patches may be insufficient to capture the fractal structure globally shown in the image, and the sequential representation of the patches leads to the loss of spatial information.

Impact of Decoder Architecture

Besides the encoder architecture, we also investigate the influence of the decoder architecture. In addition to RNN, we conduct experiments using the Transformer in [Vaswani *et al.*, 2017] as our sequential parameter decoder g_1. The results are presented in Table 3. It is observed that employing the Transformer achieves comparable inference results. How-

Encoder	Decoder	Sequence Quality↑			Image Similarity	
		BLEU	R@1	R@L	SSIM↑	LPIPS↓
Resnet50	RNN	91.40	93.34	93.33	0.9123	0.1393
	Trans.	89.76	93.26	93.01	0.9115	0.1418
Densenet	RNN	93.34	94.29	94.28	0.9179	0.1139
	Trans.	90.23	93.71	93.54	0.9144	0.1197

Table 3: Experimental results of sequential parameter estimation for different decoder architectures.



(a) MAE of Re(c) (b) MAE of Im(c) (c) MAE of N

(d) MAE of τ (e) SSIM (f) LPIPS

Figure 6: The curves of different evaluation metrics with respect to the size of training set on the Julia Set.

ever, we still apply RNN as the default decoder architecture because it consumes fewer computational resources.

Impact of Training Data Size

Additionally, we investigate the impact of training data size on our inference results. Take the inference task of Julia set as an example. We train our model with M training images, where $M \in [2000, 8000]$, and test it on 500 images. Figure 6 shows the model performance on different evaluation measurements. As we expect, the performance of our method is improved in general with the increase of training data size. This result verifies the rationality of our model.

4.3 Experiments on Generalizability

Our method provides a data-driven solution to the inference of IFS, and its generalizability is crucial for its practical applications. To analyze the generalization power of our model, we apply it to infer IFSs from some fractal images unseen during training, including natural fractal images and the L-system fractal images generated by challenging unseen rules. The results show the potential of our method in practice and point out its current limitations and our future work.

Inferring IFSs from Natural Fractal Images

As shown in Figure 7, natural objects such as spiral aloe, trees, whirlpools, and so on, exhibit multi-scale self-similarity and can be modeled as fractals. Given such natural fractal images, we first apply Laplacian operator to extract

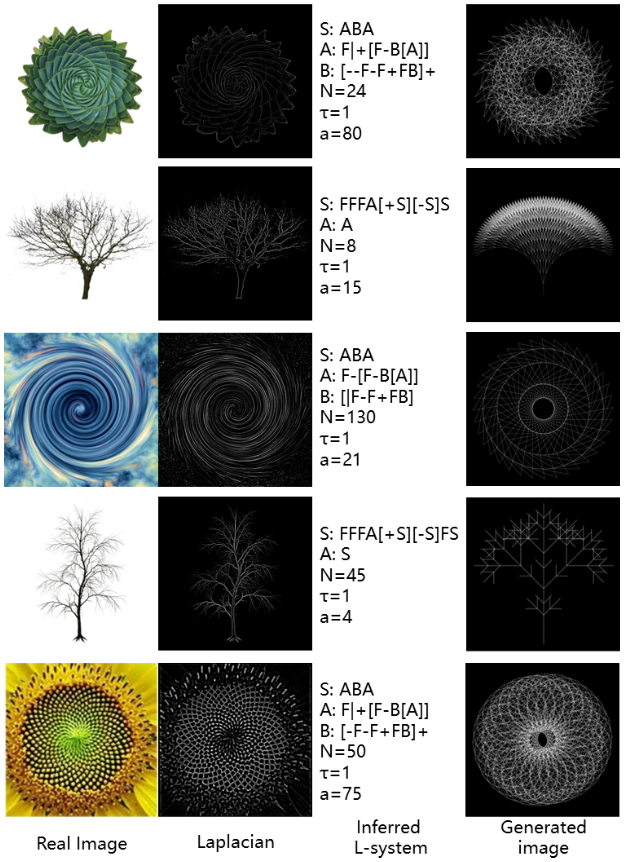


Figure 7: The result of applying our model to the natural fractal objects and their corresponding IFSs and reconstructed images.

their edge images. Then, assuming the edge images to be L-system fractal images, we pass them through our model and approximate the parameters of the corresponding L-systems accordingly. Based on the inferred L-systems, we can further generate fractal images and check their visual effects. The representative results in Figure 7 demonstrate the encouraging generalizability of our model on natural fractal images to some degree — the L-systems inferred by our model can generate fractal images that are visually similar to the original natural fractal images. These results also show the potential of our work in art design. By inferring IFSs from natural images, our model can generate artistic fractal images and provide artists with creative inspirations and materials.

Limitations on Challenging Inference Tasks

Although our model shows encouraging performance on some unseen natural fractal images, it still suffers from limited generalization power when inferring complicated IFSs. In this experiment, we create 10,000 challenging fractal images based on the L-systems with complicated rules and test our model on the dataset. Instead of considering 59 predefined rules, the rules associated with these images are randomly generated. Specifically, each image is generated by an L-system with four rules, whose vocabulary set is $\mathcal{V} = \{S, A, B, C\}$. For each rule, its predecessors are selected from $\mathcal{V}$, and its successors are sequences with three to five elements randomly sampled from $\mathcal{V} \cup \mathcal{A}$.

Encoder	Learning Paradigm	Sequence Quality↑		
		BLEU	R@1	R@L
VGG16	SL	31.34	85.88	72.17
	ZSL	24.68	86.34	74.94
Resnet50	SL	30.62	86.17	73.98
	ZSL	24.41	84.88	73.97
DenseNet	SL	32.33	84.79	72.61
	ZSL	24.03	85.91	74.85
ViT	SL	33.12	87.66	70.75
	ZSL	25.60	87.96	74.61

Table 4: Testing results on the challenging L-system fractal images.

Due to the randomness, inferring the L-systems from their images is much more challenging. When training our model, we consider two learning paradigms. The first is classic supervised learning (SL), i.e., learning the model directly based on the challenging dataset. The second is learning the model based on the original (simple) dataset and testing the model on the challenging dataset, leading to the zero-shot learning (ZSL) paradigm. The results are shown in Table 4. Compared to the results in Table 2, our model suffers from significant performance degradation in this experiment. In particular, both our training L-system images and the above natural fractal images yield relatively-simple rules, while the rules applied to generate the challenging dataset are much more complicated. As a result, our current model shows undesired ZSL performance. Additionally, the results of supervised learning are not good enough either, which means that learning a generalizable inference model for complicated IFSs requires much more training data and new learning paradigms, e.g., large-scale pre-training, which is left as our future work.

5 Conclusion

In this work, we learn a multi-head auto-encoding model to infer typical IFSs approximately based on fractal images. The proposed model leverages two decoding heads to infer sequential and non-sequential parameters of different IFSs and considers one more image decoding head to reconstruct input fractal images. We design a semi-supervised learning paradigm to learn the proposed model, making unlabeled fractal images available during training. Our method provides a promising solution to infer Julia Set and L-system approximately from fractal images. In the future, we plan to further improve the generalizability of our model by i) considering more heterogeneous IFSs and their fractal images in the training phase and ii) applying cutting-edge pre-training techniques to train powerful large-scale models.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (No. 92270110, 62106271, and 62102031), the foundation of Engineering Research Center of Next-Generation Intelligent Search and Recommendation, and the foundation of Key Laboratory of Artificial Intelligence, Ministry of Education, P.R. China.

References

[Angeline, 1994] Peter J. Angeline. Genetic programming: On the programming of computers by means of natural selection. *Biosystems*, page 69–73, Jan 1994.

[Barnsley and Vince, 2011] Michael F Barnsley and Andrew Vince. The chaos game on a general iterated function system. *Ergodic theory and dynamical systems*, 31(4):1073–1079, 2011.

[Barnsley, 1988] Michael Barnsley. *Fractals everywhere*. Academic Press Professional, Inc., 1988.

[Barry *et al.*, 2009] David Barry, Onwuarolu Ifeyinwa, Shauna McGee, Raymond Ryan, Gwilym Williams, and Jonathan Blackledge. Relating fractal dimension to branching behaviour in filamentous microorganisms. *ISAST Transactions on Electronics and Signal Processing*, 4(1):71–76, 2009.

[Bernard and McQuillan, 2023] Jason Bernard and Ian McQuillan. Stochastic l-system inference from multiple string sequence inputs. *Soft Computing*, 27(10):6783–6798, May 2023.

[de la Higuera, 2005] Colin de la Higuera. A bibliographical study of grammatical inference. *Pattern Recognition*, 38(9):1332–1348, Sep 2005.

[Dosovitskiy *et al.*, 2020] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

[Goodfellow *et al.*, 2014] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[Herman and Walker, 1972] GT Herman and A Walker. The syntactic inference problem as applied to biological systems. *Machine Intelligence*, 7:341–356, 1972.

[Hoskins and Vagners, 1992] Douglas A Hoskins and Juris Vagners. Image compression using iterated function systems and revolutionary programming: image compression without image metrics. In *Conference record of the Twenty-Sixth Asilomar Conference on Signals, Systems & Computers*, pages 705–706. IEEE Computer Society, 1992.

[Howard *et al.*, 2017] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[Huang *et al.*, 2017] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[Hussein *et al.*, 1999] HusseinKaram Hussein, Aboul-Ella Hassanien, and M. Nakajima. Escape-time modified algorithm for generating fractal images based on petri net reachability. *IEICE Transactions on Information and Systems,IEICE Transactions on Information and Systems*, Jul 1999.

[Jacquin, 1992] A.E. Jacquin. Image coding based on a fractal theory of iterated contractive image transformations. *IEEE Transactions on Image Processing*, 1(1):18–30, Jan 1992.

[Julia, 1918] Gaston Julia. Mémoire sur l'itération des fonctions rationnelles. *Journal de mathématiques pures et appliquées*, 1:47–245, 1918.

[Jürgensen and Lindenmayer, 1987] Helmut Jürgensen and Aristid Lindenmayer. Inference algorithms for developmental systems with cell lineages. *Bulletin of Mathematical Biology*, 49(1):93–123, 1987.

[Kapoor *et al.*, 2004] A. Kapoor, K. Arora, A. Jain, and G.P. Kapoor. Stochastic image compression using fractals. In *Proceedings ITCC 2003. International Conference on Information Technology: Coding and Computing*, May 2004.

[Kataoka *et al.*, 2022] Hirokatsu Kataoka, Kazushige Okayasu, Asato Matsumoto, Eisuke Yamagata, Ryosuke Yamada, Nakamasa Inoue, Akio Nakamura, and Yutaka Satoh. Pre-training without natural images. *International Journal of Computer Vision (IJCV)*, 2022.

[Kim *et al.*, 2022] Dongjun Kim, Yeongmin Kim, Se Jung Kwon, Wanmo Kang, and Il-Chul Moon. Refining generative process with discriminator guidance in score-based diffusion models. *arXiv preprint arXiv:2211.17091*, 2022.

[Kingma and Ba, 2014] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[Krizhevsky *et al.*, 2012] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[Lee, 2022] Jaegeol Lee. The crossover study regarding the idea of self-organization in organicism by immanuel kant and fractal theory: The aesthetic premise on organisms, life, and infinity. *The Korean Society of Culture and Convergence*, 44(9):419–433, Sep 2022.

[Li *et al.*, 2021] Wei Li, Can Gao, Guocheng Niu, Xinyan Xiao, Hao Liu, Jiachen Liu, Hua Wu, and Haifeng Wang. Unimo: Towards unified-modal understanding and generation via cross-modal contrastive learning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Jan 2021.

[Li *et al.*, 2022] Chenliang Li, Haiyang Xu, Junfeng Tian, Wei Wang, Ming Yan, Bin Bi, Jiabo Ye, He Chen, Guohai Xu, Zheng Cao, et al. mplug: Effective and efficient vision-language learning by cross-modal skip-connections. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7241–7259, 2022.

[Li *et al.*, 2023] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023.

[Lin, 2004] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

[Lindenmayer, 1968] Aristid Lindenmayer. Mathematical models for cellular interactions in development i. filaments with one-sided inputs. *Journal of theoretical biology*, 18(3):280–299, 1968.

[Maaten and Hinton, 2008] Laurensvander Maaten and GeoffreyE. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research,Journal of Machine Learning Research*, Jan 2008.

[Mai *et al.*, 2020] Florian Mai, Nikolaos Pappas, Ivan Montero, Noah A. Smith, and James Henderson. Plug and play autoencoders for conditional text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Jan 2020.

[Mandelbrot and Aizenman, 1979] B. B. Mandelbrot and Michael Aizenman. Fractals: Form, chance, and dimension. *Physics Today*, 32(5):65–66, May 1979.

[Mandelbrot and Mandelbrot, 1982] Benoit B Mandelbrot and Benoit B Mandelbrot. *The fractal geometry of nature*, volume 1. WH freeman New York, 1982.

[Menassel *et al.*, 2020] Rafik Menassel, Idriss Gaba, and Khalil Titi. Introducing bat inspired algorithm to improve fractal image compression. *International Journal of Computers and Applications*, 42(7):697–704, Oct 2020.

[Oppenheimer, 1986] Peter E Oppenheimer. Real time design and animation of fractal plants and trees. *ACM SiGGRAPH Computer Graphics*, 20(4):55–64, 1986.

[Papineni *et al.*, 2002] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[Pentland, 1984] Alex P Pentland. Fractal-based description of natural scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):661–674, 1984.

[Radford *et al.*, 2018] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

[Raffel *et al.*, 2020] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

[Ramesh *et al.*, 2022] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

[Rinaldo and Zakhor, 1994] R. Rinaldo and A. Zakhor. Inverse and approximation problem for two-dimensional fractal sets. *IEEE Transactions on Image Processing*, 3(6):802–820, Jan 1994.

[Robbins and Monro, 1951] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, page 400–407, 1951.

[Simonyan and Zisserman, 2015] K Simonyan and A Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[Tsai and Mecholsky, 1991] YL Tsai and JJ Mecholsky. Fractal fracture of single crystal silicon. *Journal of materials research*, 6(6):1248–1263, 1991.

[Van den Oord *et al.*, 2016] Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. *Advances in neural information processing systems*, 29, 2016.

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

[Wang *et al.*, 2004] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[Xu *et al.*, 2015] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057. PMLR, 2015.

[Zhang *et al.*, 2018] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[Zhao *et al.*, 2018] Yang Zhao, Zhaoqi Ye, Lei Wang, Hongbin Zhang, Fangqi Xue, Songhai Xie, Xiao-Ming Cao, Yahong Zhang, and Yi Tang. Engineering fractal mtw zeolite mesocrystal: Particle-based dendritic growth via twinning-plane induced crystallization. *Crystal Growth & Design*, 18(2):1101–1108, Feb 2018.