Inferring Iterated Function Systems Approximately from Fractal Images

Haotian Liu1,2*, Dixin Luo2,3, Hongteng Xu1,4†

1Gaoling School of Artificial Intelligence, Renmin University of China, Beijing
2School of Computer Science and Technology, Beijing Institute of Technology, Beijing
3Key Laboratory of Artificial Intelligence, Ministry of Education, Shanghai
4Beijing 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 [Angelina, 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.

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
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; Menassell 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:

\[ z_{n+1} = z_n^2 + c, \quad n = 0, 1, 2, \ldots, \]

where the offset \( c \in \mathbb{C} \) is a complex number. \( z_0 \) is the initial input, which is fixed as \( 0+0i \) 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
fractal image of Julia set, denoted as $I^{(J)}$, is associated with two parameters $y = \{\text{Re}(c), \text{Im}(c)\}$, where $c \in \mathbb{C}$, $\text{Re}(c)$ and $\text{Im}(c)$ 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. $\omega$ 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 $I^{(L)}$, 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 $s$, whose vocabulary set is $\mathcal{V} \cup \mathcal{A}$.

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

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F$</td>
<td>Draw forwards</td>
</tr>
<tr>
<td>$[,]$</td>
<td>Push / pop state</td>
</tr>
<tr>
<td>$+,-$</td>
<td>Rotate by +/- angle</td>
</tr>
<tr>
<td>$!$</td>
<td>Negate angle</td>
</tr>
<tr>
<td>$</td>
<td>$</td>
</tr>
</tbody>
</table>

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 $\alpha \in [0^\circ, 360^\circ]$ 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 $f: \mathcal{Z} \mapsto \mathcal{Z}$, maps fractal images to a $d$-dimensional latent space $\mathcal{Z} \subset \mathbb{R}^d$, i.e., $z = f(I)$. Based on the latent representations of fractal images, the first decoder $g_1$ predicts the sequential rules of the L-system, incorporating both the parameter 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.

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., $D_J = \{I^{(J)}, y^{(J)}\}$ and $D_L = \{I^{(L)}, s^{(L)}\}$, where $y^{(J)}$ and $s^{(L)}$ are non-sequential parameters of the Julia Set and $s^{(L)}$ represents the sequential parameters of L-system. Additionally, we denote the unlabeled fractal images we collected as $D_U = \{I^{(U)}\}$. 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:

$$L_1(f, g_1) := \sum_{(I, s) \in D_L} \sum_{s_i \in s} CE(s_i, g_1(f(I), s_i)),$$
where $s_i = \{s_j\}_{j=1}^{i-1}$ 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.,

$$L_2(f, g_2) := \sum_{(I, y) \in D_J} \|g_2(f(I)) - y\|_2^2. \tag{3}$$

**Fractal Image Reconstruction**

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

$$L_3(f, g_3) := \sum_{\mathbf{I} \in D_J \cup D_L \cup D_U} \|g_3(f(\mathbf{I})) - \mathbf{I}\|_F^2, \tag{4}$$

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 $V = \{S, A, B, C, D, E\}$. 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 $4 \times 10^{-4}$ 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 $L_3$...
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 modalities, which reflects the inherent difference between Julia and L-system fractals.
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.

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

Table 3: Experimental results of sequential parameter estimation for different decoder architectures.
Figure 7: The result of applying our model to the natural fractal objects and their corresponding IFSs and reconstructed 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.

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

<table>
<thead>
<tr>
<th>Encoder</th>
<th>Learning Paradigm</th>
<th>BLEU</th>
<th>R@1</th>
<th>R@3</th>
<th>R@5</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGG16</td>
<td>SL</td>
<td>31.34</td>
<td>85.88</td>
<td>72.17</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ZSL</td>
<td>24.68</td>
<td>86.34</td>
<td>74.94</td>
<td></td>
</tr>
<tr>
<td>Resnet50</td>
<td>SL</td>
<td>30.62</td>
<td>86.17</td>
<td>73.98</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ZSL</td>
<td>24.41</td>
<td>84.88</td>
<td>73.97</td>
<td></td>
</tr>
<tr>
<td>DenseNet</td>
<td>SL</td>
<td>32.33</td>
<td>84.79</td>
<td>72.61</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ZSL</td>
<td>24.03</td>
<td>85.91</td>
<td>74.85</td>
<td></td>
</tr>
<tr>
<td>ViT</td>
<td>SL</td>
<td>33.12</td>
<td>87.66</td>
<td>70.75</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ZSL</td>
<td>25.60</td>
<td>87.96</td>
<td>74.61</td>
<td></td>
</tr>
</tbody>
</table>

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-systems approximately from fractal images. In the future, we plan to further improve the generalizability of our model by considering more heterogeneous IFSs and their fractal images in the training phase and 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


[Julia, 1918] Gaston Julia. M´emoire sur l’it´eration des fonc-


