Learning from the Tangram to Solve Mini Visual Tasks

Yizhou Zhao[1], Liang Qiu[1], Pan Lu[1], Feng Shi[1], Tian Han[2], Song-Chun Zhu[1]

[1]UCLA Center for Vision, Cognition, Learning, and Autonomy
[2] Stevens Institute of Technology
yizhouzhao@g.ucla.edu

Abstract

Current pre-training methods in computer vision focus on natural images in the daily-life context. However, abstract diagrams such as icons and symbols are common and important in the real world. This work is inspired by Tangram, a game that requires replicating an abstract pattern from seven dissected shapes. By recording human experience in solving tangram puzzles, we present the Tangram dataset and show that a pre-trained neural model on the Tangram helps solve some mini visual tasks based on low-resolution vision. Extensive experiments demonstrate that our proposed method generates intelligent solutions for aesthetic tasks such as folding clothes and evaluating room layouts. The pre-trained feature extractor can facilitate the convergence of few-shot learning tasks on human handwriting and improve the accuracy in identifying icons by their contours. The Tangram dataset is available at https://github.com/yizhouzhao/Tangram.

Introduction

As many vision tasks are relevant, one would expect a model, particularly pre-trained from one dataset, to assist a different challenge. Traditionally, supervised pre-training on image classification has been employed to help object detection (Shinya, Simo-Serra, and Suzuki 2019) and semantic parsing (Orsic et al. 2019). Moreover, popular unsupervised pre-training has recently produced remarkable results in visual tasks such as image classification (Chen et al. 2020a) and clustering (Chakraborty, Gosthipaty, and Paul 2020). The common datasets to train basic models include PASCAL VOC (Everingham et al. 2010), ImageNet (Deng et al. 2009), and COCO (Lin et al. 2014), all of which contain photographs.

It is natural to start the pre-training process from real-life images to solve daily vision tasks. However, one of the underlying limitations of current works is their focus on content from natural images. Besides natural images, abstract diagrams, such as texts, symbols, and signs, also carry rich visual semantics and account for a large part of the visual world. For instance, it is shown that emojis can express rich human sentiments (Felbo et al. 2017), and diagrams like icons can map the physical worlds into symbolic and aesthetic representations (Lagunas, Garces, and Gutierrez 2019; Madan et al. 2018; Karamatsu et al. 2020). Furthermore, most of the tasks related to natural images can be accomplished by low-resolution vision (Land and Nilsson 2012) (see Figure 2). Therefore, training an enormous backbone (e.g., a deep residual network (He et al. 2016)) to solve tasks related to abstract diagrams complicates the problem.

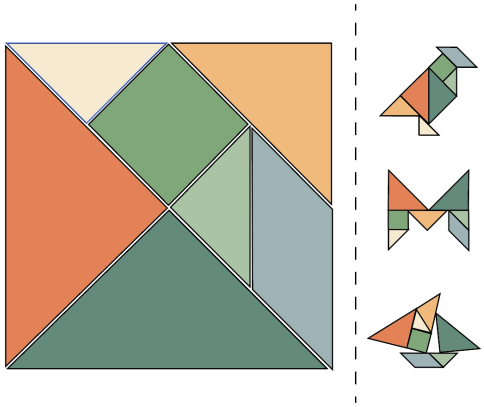

Figure 1: The left panel shows the square representation of the Tangram that consists of five triangles of three sizes, one parallelogram and one square. The right panel shows some tangram puzzles: a bird, the letter M and a sailboat.

In this paper, we argue that we can solve the tasks related to abstract diagrams by learning from the process of replicating a tangram puzzle. The tangram, a dissection puzzle consisting of seven planar polygons (tans), is world-famous and has been used for many purposes, including art, design, and education. Although it only consists of seven tans, it can generate thousands of meaningful patterns such as animals, buildings, letters, and numbers. Solving a tangram puzzle associates with our cognitive and imaginative abilities.

We introduce the **Tangram**, a new dataset consisting of more than $10,000$ snapshots recording the steps to solve a total number of 388 tangram puzzles. A neural model can be pre-trained from the Tangram to solve two groups of downstream tasks.

The first group is about aesthetics. We introduce two toy tasks: folding clothes and organizing furniture (room layouts). Tuning the pre-trained network from several expert

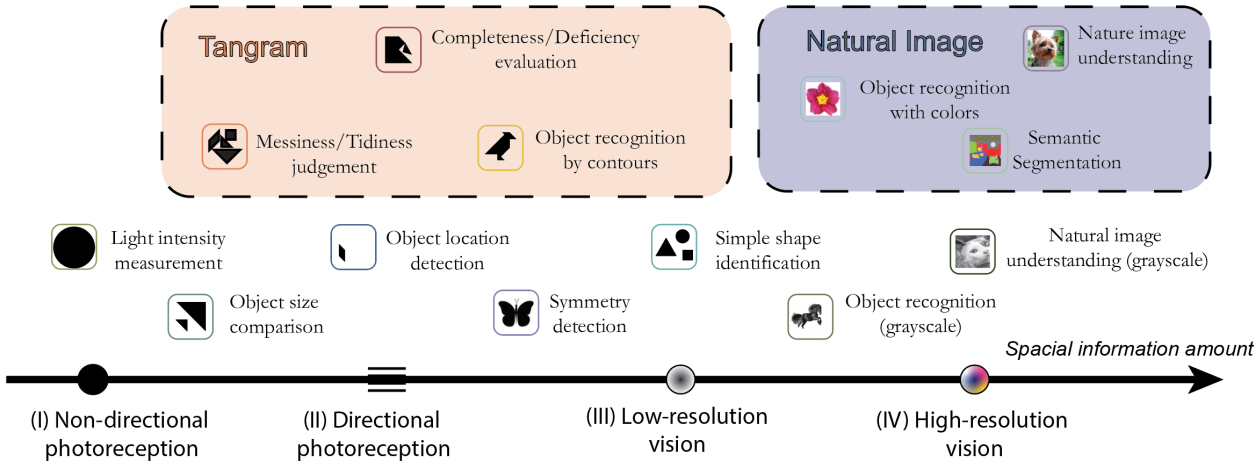

Figure 2: Visual perception tasks ranked by the amount of spatial information. In biology, visual perception tasks are divided into four levels based on the number of photoreceptors (Land and Nilsson 2012). Our Tangram dataset relates to many low-resolution visual tasks, while current works usually focus on high-resolution natural images.

samples can generate an aesthetic landscape that helps make aesthetic judgments. Experiments show that our method performs best when cooperating with max-entropy inverse reinforcement learning (Ziebart et al. 2008) and generative adversarial imitation learning (Ho and Ermon 2016).

The second group includes several recognition tasks. In the N-way-K-shot setting, we show that conducting pre-training on the Tangram improves the performance of recognizing the human handwriting, including Omniglot (Lake, Salakhutdinov, and Tenenbaum 2019) and Multi-digit MNIST (Sitzmann et al. 2020). This method also improves the performance of icon recognition from contours.

This paper makes three major contributions:

- To our best knowledge, by introducing Tangram, we are the pioneers to suggest applying transfer learning from the human gaming experience to solve vision tasks.
- We demonstrate that pre-training from the Tangram can help solve both low-level aesthetics tasks and recognition tasks.
- We show that pretraining on the Tangram facilitates convergence in few-shot learning tasks, and improves the performance of recognition under low-level vision.

Related Work

An abundance of related work inspires our work, including pre-training in computer vision, rating image aesthetics with deep learning, and few-shot learning.

Pre-training

Pre-training methods can be either supervised or unsupervised. The supervised pre-training on ImageNet is conventional for object recognition, localization, and segmentation (He, Girshick, and Dollár 2019). Inspired by the success of unsupervised pre-training in natural language processing, the community has gained much interest in studying unsupervised pre-training in computer vision, such as contrastive training (Chen et al. 2020b), self-supervised training (Jing and Tian 2020). In many tasks, fine-tuning from a pre-trained model is faster than training from scratch. Pre-training can also help when high-quality labeled data is scarce.

Image Aesthetics

Image aesthetics assessment attempts to quantify an image's beauty. Image quality is influenced by numerous factors such as color (Nishiyama et al. 2011), lighting (Freeman 2007), texture (Ke, Tang, and Jing 2006), and image composition (Deng, Loy, and Tang 2017). While subjective judgment by human eyes is the most reliable way to evaluate image quality, the beauty of an image can also be assessed by well-established photographic theories (Zhai and Min 2020). Recent research has shown that data-driven approaches can be more efficient, especially those that employ feature extraction by multi-column convolutional neural networks (CNNs) (Lu et al. 2015; Doshi, Shikkenawis, and Mitra 2019). Popular databases for image quality assessment (IQA) are mainly collected as photos (natural images), such as the Photo.Net database (Joshi et al. 2011) and the CUHK-PhotoQuality database (Luo, Wang, and Tang 2011). Some emerging databases consist of images from virtual contents such as screen content image quality database (SCIQ) (Ni et al. 2017) and compressed Virtual reality image quality database (CVIQ) (Sun et al. 2019b).

Few-shot Learning

The main goal of few-shot learning is to learn new tasks with a few support examples while maintaining the ability to generalize. Recently, there has been a growing interest in achieving the goal by learning prior knowledge from previous tasks, especially training feature extractors that can efficiently segregate novel classes (Hu, Gripon, and Pateux

2020).

We apply our Tangram dataset to train the feature-extracting parts of optimization-based meta-learning algorithms such as MAML (Finn, Abbeel, and Levine 2017) and ANIL (Raghu et al. 2019). Besides, since the Tangram only contains shapes and contours, we perform experiments on the few-shot learning tasks that are color-free and texture-free, for example, the Omniglot challenge (Lake, Salakhutdinov, and Tenenbaum 2019).

Pre-training from the Tangram

Data Collection

To collect the process of solving puzzles from human experience, an interactive labeling tool is developed using the Unity game engine (Haas 2014). The labeling tool can record every step of moving, rotating, or flipping of one tan as a snapshot. Seven lab technicians spent weeks on completing a total number of 776 solutions to 388 unique puzzles, capturing more than $10,000$ snapshots.

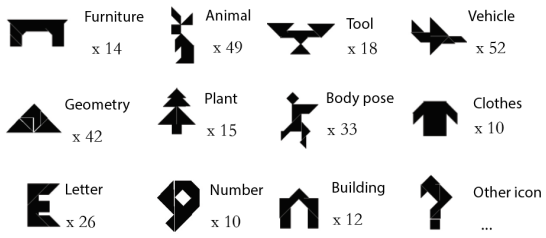

Figure 3: Collected examples of different categories in the Tangram dataset.

Figure 3 illustrates an overview of the puzzles types and their counts. The Tangram dataset consists of diverse tangram patterns including animals, plants, letters, numbers, buildings, human poses, and some everyday objects. It requires necessary perceptive recognition and elementary geometry skills to solve them. We will release the dataset to the public to encourage further study into abstract image understanding.

Learning from Puzzles

Denote the order set $(I_1, I_2, ..., I_{n_p})$ as the process to solve a tangram puzzle P, where each $I_i, i \in \{1, ..., n_p\}$ is an image representing one step toward the solution, and n_p is the total number of steps. Since a tangram pattern only has shapes and contours, I_i is a binary image with size $H \times W$.

What can we learn from the puzzles, and how can we use the solving steps? We argue that the Tangram reveals two pieces of information:

- The step-by-step solving process leads to more complete and tidy shape combinations, containing the perception of beauty.
- There is a connection between the pattern and the name of the object due to correspondence between the final completed pattern and a real-world object.

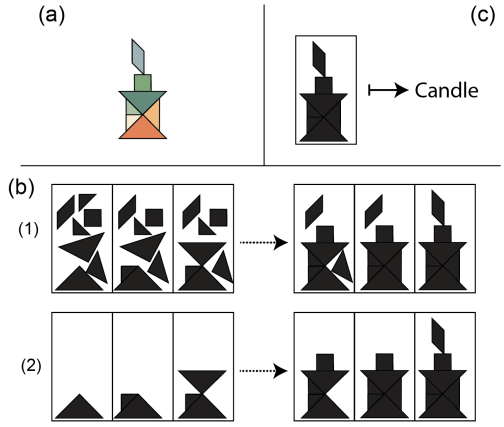

Figure 4: (a) The expected solution of a tangram puzzle. (b) The process of solving the puzzle with its two variants. (c) The final completed puzzle image and the meaning of the item.

Therefore, we formulate two learning goals and assign two loss functions.

Let $f_\theta : \{0,1\}^{H\times W} \mapsto [0,1]$ be the function indicating the degree of completeness of step I_i. We define the **completeness contrast loss** (CCL) for the process $(I_i)_{i=1}^{n_p}$ as

$$\begin{aligned}\mathrm{CCL}(I_1, ..., I_p) = &(0 - f_\theta(I_1))^2 \\ &+ \sum_{t=1}^{n_p-1} \Big(f_\theta(I_t) - f_\theta(I_{t+1})\Big)^2 \\ &+ (f_\theta(I_{n_p}) - 1)^2. \end{aligned} \tag{1}$$

By the Cauchy–Schwarz inequality, CCL reaches minimum value $\frac{1}{n_p+1}$ when $f_\theta(I_i) = i/(n_p+1), i = 1, 2, ..., n_p$. Minimizing CCL results in a right order for $(I_i)_{i=1}^{n_p}$.

Let $g_\phi : \{0,1\}^{H\times W} \mapsto \mathbb{R}^{N_{\text{word}}}$ map the binary image to the word embedding W_P of a pattern P, where N_{word} is the dimension of the embedding space. The **puzzle meaning loss** (PML) for the final step I_{n_p} is defined as

$$\mathrm{PML}(I_{n_p}) = |g_\phi(I_{n_p}) - W_P|^2. \tag{2}$$

Figure 4 depicts an implementation of the two loss functions described above. Panel (b) demonstrates two variants of the puzzle-solving processes. The first variant traces all tans, recording progression from disorganization to neatness; the second variant traces only the final state of moved tans and represents a progression from fragmentation to completeness.

To train the functions f_θ and g_ϕ, we use a simple convolutional neural network with only four 3×3 convolutional layers. Each image is resized into 28×28. We apply the 50-dimension GloVe embedding (Pennington, Socher, and Manning 2014) for pattern names, and we assign 80% of the weight on CCL and 20% on PML. The feature extraction part of the network is transferred to achieve other challenges.

Experiments

We define **mini visual tasks** as the vision tasks that only require learning from low-resolution binary images. We divide mini visual tasks into two categories: aesthetic tasks and recognition tasks. We choose folding clothes and generating room layouts (organizing furniture) as representatives for the first category, identifying human hand-writings and recognizing icons for the second.

Folding Clothes

Folding clothes is a classic task in robotics that has received heated discussion among various works. Prevalent methods include grounding human demonstration from videos (Yang et al. 2015), employing random decision forests and probabilistic planning (Doumanoglou et al. 2014), using deep reinforcement learning (Jangir, Alenyà, and Torras 2020), and designing a modifiable stochastic grammar (Xiong et al. 2016).

We abstract the clothes-folding challenge as a purely visual task: the contour of the dress/suit/shirt/pants is represented by a binary image, and folding clothes is characterised by manipulating images. Figure 5 shows an image-like abstraction of folding a dress.

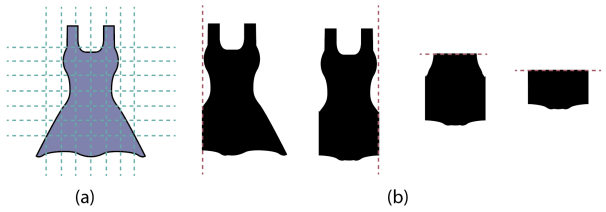

Figure 5: (a) A dress with folding axes. (b) Folding steps.

The current state of the clothes s is represented by a binary image I from image space $\mathcal{S} = \{0,1\}^{H\times W}$, and an action a leads to fold the image along a certain axis (see figure 5). We also regard this task as a few-shot learning problem: as we are only given a few expert trajectories $\pi_E = \{\tau_{E_1}, \tau_{E_2}, ..., \tau_{E_{n_e}}\}$, where each trajectory τ_{E_i} is represented by the order sequence of states $(s_{E_{i1}}, s_{E_{i2}}, ...)$ towards the solution, the problem is how we can fold other arbitrary clothes we have not seen before.

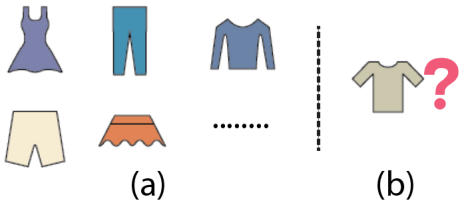

Figure 6: (a) Expert sample clothes. (b) A T-shirt unseen before.

We try several different ways to solve this task, including directly minimizing the CCL for expert trajectories and drawing on the popular algorithms from inverse reinforcement learning (IRL). The algorithms listed below can be applied not only to perform clothes-folding and furniture-organizing, but to solve a wide range of challenges related to robotics.

- **Score learning** (SL): we can direclty give a score to a state $V_\delta : \mathcal{S} \mapsto [0,1]$, by learning from expert trajectories with the CCL (see equation 1):

$$V_\delta(s) := f_\theta(s). \tag{3}$$

- **Max-entropy inverse reinforcement learning** (ME-IRL) (Ziebart et al. 2008): suppose a trajectory $\tau_i = (s_1, s_2, ...)$ is sampled from the current cloth-folding policy π_i, and $F_\psi : \mathcal{S} \mapsto [0,1]$, is the evaluation function for state s, we can calculate the gradient of ψ by

$$\frac{\partial \mathcal{L}_\psi}{\partial \psi} = \mathbb{E}_{s\sim\tau_E}\left[\frac{\partial F_\psi(s)}{\partial \psi}\right] - \mathbb{E}_{s\sim\tau_i}\left[\frac{\partial F_\psi(s)}{\partial \psi}\right], \tag{4}$$

where $\mathcal{L}_\psi = P(\tau|\pi_i, \tau \in \pi_E)$ is the likelihood function of taking expert trajectories under the current policy.

- **Generative adversarial imitation learning** (GAIL) (Ho and Ermon 2016): after initializing the discriminator function $D_\omega : \mathcal{S} \mapsto [0,1]$ to distinguish states between expert and sampling trajectories, we can update ω with gradient

$$\begin{aligned}\frac{\partial \mathcal{L}_\omega}{\partial \omega} &= \mathbb{E}_{s\sim\tau_E}\left[\frac{\partial \log D_\omega(s)}{\partial \omega}\right] \\ &+ \mathbb{E}_{s\sim\tau_i}\left[\frac{\partial \log(1 - D_\omega(s))}{\partial \omega}\right]\end{aligned} \tag{5}$$

where $\mathcal{L}_\omega$ is the adversarial loss (Ho and Ermon 2016) and τ_i shares the same meaning as above. Notice that we make a modification to GAIL by only distinguishing the state s instead of the state-action pair (s, a) since we are not given enough state-action pairs under few-shot settings.

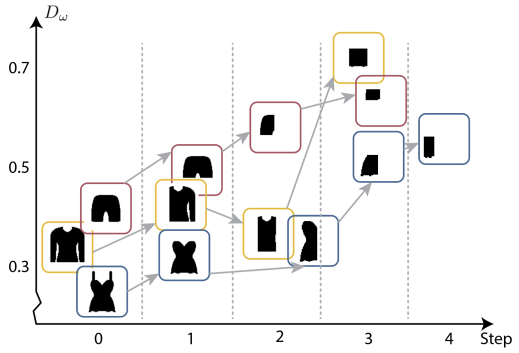

Figure 7: Aesthetic scores induced by D_w (pre-trained).

For simplicity, we regard the greedy policy deduced by the value of V_δ, F_ϕ and D_ω as the propagated policy π_i for SL, ME-IRL and GAIL. We assume that the clothes are put straight initially and they can only be folded along vertical

and horizontal axes. The size of the image I representing the state s is 28×28 and there are ten vertical and ten horizontal folding axes evenly distributed in the image.

We apply the network of the same structure in Section *Pre-training from the Tangram* for feature extraction to calculate V_δ, F_ψ and D_ω. Three different ways along with pre-training or non-pre-training cases provide us with six different models. The models are trained on the expert trajectories from a total number of 18 clothes, including dresses, long shirts, T-shirts, trousers, short pants, and skirts (three for each type). Then, models are tested on six new clothes from the aforementioned types and three clothes from other types.

We refer to V_δ, F_ϕ and D_ω derived from equations 3, 4, and 5 as the *aesthetic scores* of cloth-folding. Figure 7 illustrates that D_ω increases as the clothes-folding process goes along. We compare the performance between different models by calculating the ranking of the ordered states $(s_{E_{i1}}, s_{E_{i2}}, ...)$ of expert trajectories based on V_δ, F_ϕ and D_ω. Since on average the length of expert trajectories is around four, we only consider the precision at K (P@K) with $K \leq 3$. Recall at K as gives similar results.

Table 1 compares the overall difference in P@K between the pre-trained model and the non-pre-trained model (training from scratch) for the training expert samples (see the detailed comparison for each model in the Appendix). In general, we can see that pre-training improves the training precision and reduces the variance. We select the best models of the six methods and test them once on the nice clothes that are unseen before. Table 2 shows the mean and standard deviation of testing P@K. Except that ME-IRL without pre-training outperforms the pre-trained one w.r.t. P@1, pre-training improves the overall test accuracy, and the high precision on each value ($K = 1, 2, 3$) implicates overall better aesthetic scores.

ME-IRL and GAIL are common data-driven algorithms in the IRL domain. As with SL, their performance is heavily dependent on the amount of expert data given for training. Therefore, tuning from a pre-trained model can alleviate data reliance.

	P@1	P@2	P@3
From scratch	0.54 ± 0.5	0.66 ± 0.3	0.76 ± 0.2
Pre-training	0.77 ± 0.4	0.84 ± 0.3	0.86 ± 0.2

Table 1: The mean and standard deviation of training P@K: a comparison between models with or without pre-training.

	P@1	P@2	P@3
SL	0.22 ± 0.46	0.44 ± 0.46	0.55 ± 0.47
+ Pre	$\mathbf{0.89 \pm 0.33}$	0.78 ± 0.26	0.81 ± 0.18
ME-IRL	$\mathbf{0.89 \pm 0.33}$	0.78 ± 0.26	0.74 ± 0.22
+ Pre	0.67 ± 0.50	$\mathbf{0.94 \pm 0.17}$	0.96 ± 0.11
GAIL	0.33 ± 0.25	0.61 ± 0.33	0.74 ± 0.22
+ Pre	$\mathbf{0.89 \pm 0.33}$	$\mathbf{0.94 \pm 0.17}$	$\mathbf{1.00 \pm 0.00}$

Table 2: The mean and standard deviation of testing P@K.

Evaluating Room Layouts

Generating room layouts is different from folding clothes in that the latter focuses on the shape change of a single object, while the former requires arranging multiple objects. These two pre-training exercises may correspond to the two variants of a replicating process of a tangram puzzle(see figure 4).

The study of the layout generation has been active in various domains such as architectural design (Nauata et al. 2020; Bao et al. 2013) and game level design (Ma et al. 2014; Hendrikx et al. 2013). We focus on the task of generating content for indoor scenes, especially furniture arrangement (Yu et al. 2011; Ritchie, Wang, and Lin 2019; Qi et al. 2018), and abstract it as a purely visual task as shown in Figure 8.

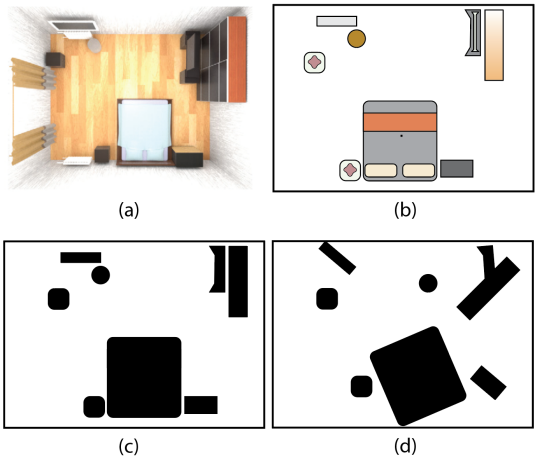

Figure 8: (a) Original indoor scene sample from (Qi et al. 2018). (b) Abstract room layout. (c) Binary image representation. (d) Room messed up.

We apply the state-of-the-art indoor scene synthesis using stochastic grammar (Qi et al. 2018) to generate the ground truth. Step by step, we perturb the room layout by the action a that changes the position (10 pixels each step) and angle (15° each step) of the furniture, and the reversed steps generate an expert trajectory τ_{E_i} to tidy up the room.

	P@1	P@2	P@3
From scratch	0.18 ± 0.2	0.23 ± 0.3	0.32 ± 0.3
Pre-training	0.23 ± 0.4	0.28 ± 0.4	0.39 ± 0.4

Table 3: Training P @ K comparison between models with or without pre-training.

	Original	Perturbed
GAIL (from scratch)	0.25 ± 0.45	0.23 ± 0.38
GAIL (pre-trained)	0.31 ± 0.41	0.29 ± 0.35

Table 4: Testing accuracy (P@1) of ranking the best room layout.

As in the previous experiment, we use a binary image I to represent the current state s, and apply the three functions V_δ, F_ψ and D_ω to generate the aesthetic landscapes of the room. We only train our methods from 30 generated expected trajectories and test them on 10 groups of new room organizing trajectories.

Table 3 shows the overall training improvement by pre-training. As in the previous experiment, pre-training improves the training accuracy. We select the best model GAIL from training, and we test it on identifying the best room layout from the testing trajectories. We also perturb each room in the trajectory a little to test the robustness of the model. Table 4 compares GAIL with/without pre-training on the testing challenges. The results indicate that pre-training on the Tangram improves performance in choosing the best room layout.

Few-shot Learning

The goal of few-shot learning is to utilize new data having seen only a few samples. In this section, we focus on the N-way-K-shot classification: a typical problem to discriminate between N classes with only K samples from each to train from.

The method we propose follows the paradigm of meta-learning (Sun et al. 2019a): we first train a feature extractor as a base-learner, which is later fine-tuned for another task through a meta-learner. As in previous experiments, a base learner is trained from the Tangram dataset, and then we perform a meta-test on the challenge of Omniglot (Lake, Salakhutdinov, and Tenenbaum 2019) and Multi-digit MNIST (Chen et al. 2018), where a binary image brings enough information to do classification.

We select three methods: MAML (Finn, Abbeel, and Levine 2017), ANIL (Raghu et al. 2019) and Prototypical Networks (Snell, Swersky, and Zemel 2017) to train the meta-learner from our base-learner. MAML is a popular meta-learning algorithm for few-shot learning, achieving competitive performance on several benchmark few-shot learning problems. ANIL simplifies MAML by alleviating the inner training loop but keeping the training procedure for the task-specific part. Prototypical networks learn to map the prototypes to a metric space, and then distances between prototypes and encoded query inputs are used to make the classification. To test the base-learner (feature extractor) trained on our Tangram data, we compare it with base-learners trained from EMNIST (Cohen et al. 2017) and Fashion-MNIST (Xiao, Rasul, and Vollgraf 2017)[1]. All base-learners share the same network structure.

Before moving on to fine-tuning, we compare the feature extractors obtained by training on the above datasets. We train only the last layer of the network as logistic regression. As can be seen from Table 5 and Table 6, feature extractors pre-trained on the Tangram, EMNIST, and Fashion-MNIST perform a lot better than the randomly initialized feature extractor. Except that the base-learner trained on EMNIST performs best in the 5-way-5-shot task on Omniglot, base-learners trained on the Tangram are powerful on other tasks, demonstrating their better adaptability.

[1]we did not train the base-learner on MNIST(Deng 2012) because it is highly related to Multi-digit MNIST.

	Omniglot	Double-MNIST
Random	$33.7\% \pm 2.0\%$	$7.3\% \pm 1.5\%$
EMNIST	$55.0\% \pm 5.4\%$	$26.8\% \pm 2.2\%$
Fashion-MNIST	$43.9\% \pm 4.1\%$	$30.1\% \pm 1.2\%$
Tangram	$\mathbf{56.0\% \pm 4.7\%}$	$\mathbf{36.0\% \pm 2.7\%}$

Table 5: Five-way-five-shot learning: the mean and the standard deviation of testing accuracy (logistic regression only).

	Omniglot	Double-MNIST
Random	$8.0\% \pm 0.7\%$	$6.1\% \pm 0.1\%$
EMNIST	$\mathbf{22.1\% \pm 1.2\%}$	$7.5\% \pm 0.1\%$
Fashion-MNIST	$15.6\% \pm 1.4\%$	$9.2\% \pm 0.5\%$
Tangram	$22.0\% \pm 1.0\%$	$\mathbf{10.5\% \pm 1.0\%}$

Table 6: Twenty-way-five-shot learning: the mean and the standard deviation of testing accuracy (logistic regression only).

Figure 9 compares the tuning process of different base-learners. Tuning the baser-learners pre-trained from the Tangram dataset guarantees the final performance compared with learning from scratch, while in some tasks it even speeds up convergence. However, for the other two feature extractors trained from EMNIST and FashionMNIST, although they may have a good start in some tasks, overall they tend to undermine the convergence speed and the final results, which reflects the difficulty of tuning a baser-learner for an irrelevant task. This result also demonstrates the importance of selecting a proper fundamental learning dataset in transfer learning.

Table 8 and Table 9 compare the final training results between training from scratch and pre-training from Tangram, where we apply ANIL as the tuning algorithm. The results shown are trained after 500 epochs. From the tables, we can see that pre-training from the Tangram provides slightly better results than training from scratch.

Icon Recognition

In this section, we study the recognition of abstract icons. While recognition tasks in natural pictures have been booming in the literature, visual abstraction receives comparably less attention.

At first glance, icon recognition is a relatively straightforward task compared to the recognition task in natural images, since most icons are simple shapes that are not affected by light or blocking. However, it is worth considering how these abstract icons are formed, and how these seemingly simple icons can convey a variety of meanings. In this part, we wonder whether pre-training on the Tangram dataset assists in recognition of icons. Icons-50 (Hendrycks and Dietterich 2018) is a collection with 50 types of icons and thousands of training samples. We run the experiments with Icons-50 and test our methods on Flowers-17 and Flowers-102 (Nilsback and Zisserman 2008).

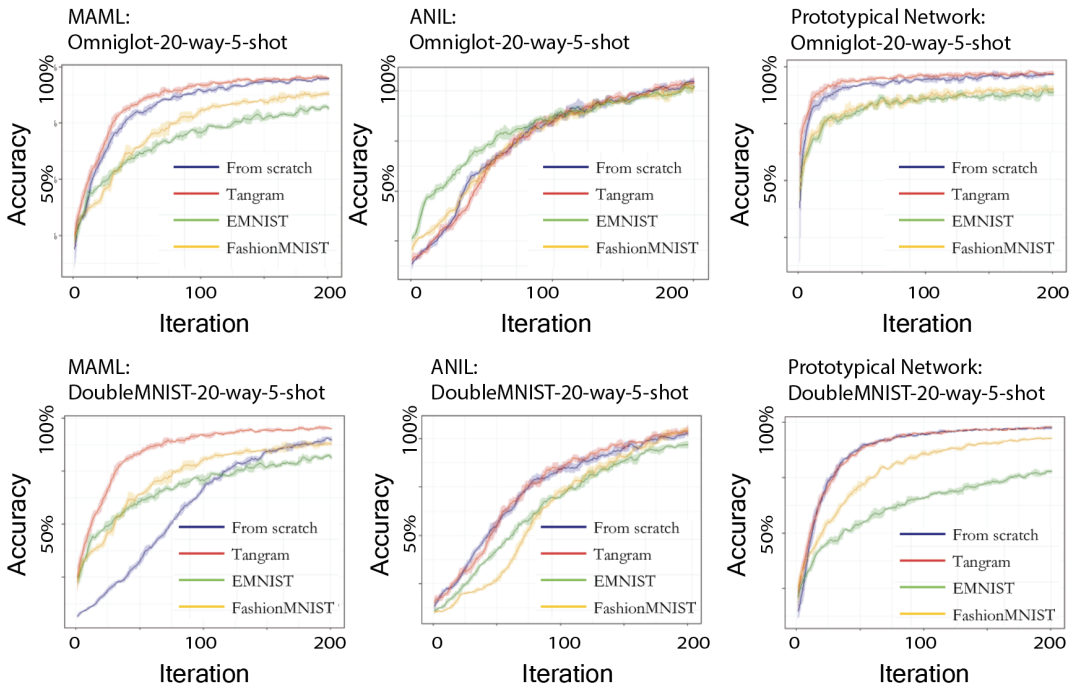

Figure 9: Testing accuracy of base-learners for different algorithms on different tasks.

	Flowers-17		**Flowers-102**		**Icons-50**	
	ResNet-18	***EfficientNet-b0***	***ResNet-18***	***EfficientNet-b0***	***ResNet-18***	***EfficientNet-b0***
From Scratch	$73.5\% \pm 3.4\%$	$76.1\% \pm 1.4\%$	$50.5\% \pm 1.3\%$	$51.7\% \pm 1.5\%$	$86.5\% \pm 0.4\%$	$84.5\% \pm 0.7\%$
Tangram	$\mathbf{76.3\% \pm 3.8\%}$	$76.0\% \pm 1.2\%$	$\mathbf{51.1\% \pm 0.8\%}$	$50.6\% \pm 1.1\%$	$\mathbf{87.1\% \pm 1.1\%}$	$85.0\% \pm 1.0\%$

Table 7: Classification results between training from scratch and pre-training from the Tangram. The inputs are binary images representing the contours only.

	Omiglot	**Double MNIST**
From scratch	$97.1\% \pm 1.4\%$	$98.4\% \pm 1.3\%$
Tangram	$98.1\% \pm 1.0\%$	$98.5\% \pm 1.0\%$

Table 8: Five-way-five-shot testing accuracy after training by ANIL.

	Omiglot	**Double MNIST**
From scratch	$92.4\% \pm 1.0\%$	$98.2\% \pm 0.3\%$
Tangram	$93.5\% \pm 0.9\%$	$98.2\% \pm 0.2\%$

Table 9: Twenty-way-five-shot testing accuracy after training by ANIL.

For Icons-50, we select icons with a white background coverage greater than 40% and draw their contours, which results in a total number of $2,450$ samples. Flowers-17 and Flower-102 are well labeled with flower contours. Flowers-17 contains 17 flower types and 849 samples, and Flowers-102 has 102 flower types and $8,189$ samples. For each dataset, 80% of the samples are used for training and the remaining 20% for testing. We use ResNet-18 (He et al. 2016) and EfficientNet-b0 (Tan and Le 2019) as the network architectures for icon classification. The inputs of the network are binary images of the size 224×224. Table 7 compares the

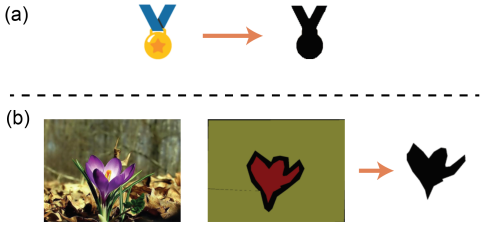

Figure 10: Data processing for (a) icons and (b) flowers.

model trained from scratch and the model pre-trained from Tangram. Although training Efficient-n0 from scratch brings good performance, the pre-trained model with ResNet-18 shows overall better testing accuracy.

Conclusion

In this paper, we introduce the Tangram dataset that records step-by-step solutions to a tangram puzzle from human experience. The pre-training on the Tangram is applied to various tasks, including folding clothes, evaluation room layouts, few-shot learning challenges, and icon classification by contours. We hope that our pioneer work in abstract visual content could inspire the community to study visual anesthetics and image abstraction.

References

Bao, F.; Yan, D.-M.; Mitra, N. J.; and Wonka, P. 2013. Generating and exploring good building layouts. *ACM Transactions on Graphics (TOG)*, 32(4): 1–10.

Chakraborty, S.; Gosthipaty, A. R.; and Paul, S. 2020. G-SimCLR: Self-Supervised Contrastive Learning with Guided Projection via Pseudo Labelling. *arXiv preprint arXiv:2009.12007*.

Chen, F.; Chen, N.; Mao, H.; and Hu, H. 2018. Assessing four neural networks on handwritten digit recognition dataset (MNIST). *arXiv preprint arXiv:1811.08278*.

Chen, M.; Radford, A.; Child, R.; Wu, J.; Jun, H.; Dhariwal, P.; Luan, D.; and Sutskever, I. 2020a. Generative pretraining from pixels. In *Proceedings of the 37th International Conference on Machine Learning*, volume 1.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020b. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*.

Cohen, G.; Afshar, S.; Tapson, J.; and Van Schaik, A. 2017. EMNIST: Extending MNIST to handwritten letters. In *2017 International Joint Conference on Neural Networks (IJCNN)*, 2921–2926. IEEE.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*.

Deng, L. 2012. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6): 141–142.

Deng, Y.; Loy, C. C.; and Tang, X. 2017. Image aesthetic assessment: An experimental survey. *IEEE Signal Processing Magazine*, 34(4): 80–106.

Doshi, N.; Shikkenawis, G.; and Mitra, S. K. 2019. Image Aesthetics Assessment Using Multi Channel Convolutional Neural Networks. In *International Conference on Computer Vision and Image Processing*, 15–24. Springer.

Doumanoglou, A.; Kargakos, A.; Kim, T.-K.; and Malassiotis, S. 2014. Autonomous active recognition and unfolding of clothes using random decision forests and probabilistic planning. In *2014 IEEE international conference on robotics and automation (ICRA)*, 987–993. IEEE.

Everingham, M.; Van Gool, L.; Williams, C. K.; Winn, J.; and Zisserman, A. 2010. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2): 303–338.

Felbo, B.; Mislove, A.; Søgaard, A.; Rahwan, I.; and Lehmann, S. 2017. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1615–1625.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *ICML*.

Freeman, M. 2007. *The complete guide to light & lighting in digital photography*. Sterling Publishing Company, Inc.

Haas, J. 2014. A history of the unity game engine. *Diss. WORCESTER POLYTECHNIC INSTITUTE*.

He, K.; Girshick, R.; and Dollár, P. 2019. Rethinking imagenet pre-training. In *Proceedings of the IEEE international conference on computer vision*, 4918–4927.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hendrikx, M.; Meijer, S.; Van Der Velden, J.; and Iosup, A. 2013. Procedural content generation for games: A survey. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 9(1): 1–22.

Hendrycks, D.; and Dietterich, T. G. 2018. Benchmarking neural network robustness to common corruptions and surface variations. *arXiv preprint arXiv:1807.01697*.

Ho, J.; and Ermon, S. 2016. Generative adversarial imitation learning. In *Advances in neural information processing systems*, 4565–4573.

Hu, Y.; Gripon, V.; and Pateux, S. 2020. Leveraging the Feature Distribution in Transfer-based Few-Shot Learning. *arXiv preprint arXiv:2006.03806*.

Jangir, R.; Alenyà, G.; and Torras, C. 2020. Dynamic Cloth Manipulation with Deep Reinforcement Learning. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 4630–4636. IEEE.

Jing, L.; and Tian, Y. 2020. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Joshi, D.; Datta, R.; Fedorovskaya, E.; Luong, Q.-T.; Wang, J. Z.; Li, J.; and Luo, J. 2011. Aesthetics and emotions in images. *IEEE Signal Processing Magazine*, 28(5): 94–115.

Karamatsu, T.; Benitez-Garcia, G.; Yanai, K.; and Uchida, S. 2020. Iconify: Converting Photographs into Icons. In *Proceedings of the 2020 Joint Workshop on Multimedia Artworks Analysis and Attractiveness Computing in Multimedia*, 7–12.

Ke, Y.; Tang, X.; and Jing, F. 2006. The design of high-level features for photo quality assessment. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, 419–426. IEEE.

Lagunas, M.; Garces, E.; and Gutierrez, D. 2019. Learning icons appearance similarity. *Multimedia Tools and Applications*, 78(8): 10733–10751.

Lake, B. M.; Salakhutdinov, R.; and Tenenbaum, J. B. 2019. The Omniglot challenge: a 3-year progress report. *Current Opinion in Behavioral Sciences*, 29: 97–104.

Land, M. F.; and Nilsson, D.-E. 2012. *Animal eyes*. Oxford University Press.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, 740–755. Springer.

Lu, X.; Lin, Z.; Jin, H.; Yang, J.; and Wang, J. Z. 2015. Rating image aesthetics using deep learning. *IEEE Transactions on Multimedia*, 17(11): 2021–2034.

Luo, W.; Wang, X.; and Tang, X. 2011. Content-based photo quality assessment. In *2011 International Conference on Computer Vision*, 2206–2213. IEEE.

Ma, C.; Vining, N.; Lefebvre, S.; and Sheffer, A. 2014. Game level layout from design specification. In *Computer Graphics Forum*, volume 33, 95–104. Wiley Online Library.

Madan, S.; Bylinskii, Z.; Tancik, M.; Recasens, A.; Zhong, K.; Alsheikh, S.; Pfister, H.; Oliva, A.; and Durand, F. 2018. Synthetically trained icon proposals for parsing and summarizing infographics. *arXiv preprint arXiv:1807.10441*.

Nauata, N.; Chang, K.-H.; Cheng, C.-Y.; Mori, G.; and Furukawa, Y. 2020. House-GAN: Relational Generative Adversarial Networks for Graph-constrained House Layout Generation. *arXiv preprint arXiv:2003.06988*.

Ni, Z.; Ma, L.; Zeng, H.; Chen, J.; Cai, C.; and Ma, K.-K. 2017. ESIM: Edge similarity for screen content image quality assessment. *IEEE Transactions on Image Processing*, 26(10): 4818–4831.

Nilsback, M.-E.; and Zisserman, A. 2008. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, 722–729. IEEE.

Nishiyama, M.; Okabe, T.; Sato, I.; and Sato, Y. 2011. Aesthetic quality classification of photographs based on color harmony. In *CVPR 2011*, 33–40. IEEE.

Orsic, M.; Kreso, I.; Bevandic, P.; and Segvic, S. 2019. In defense of pre-trained imagenet architectures for real-time semantic segmentation of road-driving images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 12607–12616.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543.

Qi, S.; Zhu, Y.; Huang, S.; Jiang, C.; and Zhu, S.-C. 2018. Human-centric indoor scene synthesis using stochastic grammar. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5899–5908.

Raghu, A.; Raghu, M.; Bengio, S.; and Vinyals, O. 2019. Rapid Learning or Feature Reuse? Towards Understanding the Effectiveness of MAML. In *International Conference on Learning Representations*.

Ritchie, D.; Wang, K.; and Lin, Y.-a. 2019. Fast and flexible indoor scene synthesis via deep convolutional generative models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6182–6190.

Shinya, Y.; Simo-Serra, E.; and Suzuki, T. 2019. Understanding the Effects of Pre-Training for Object Detectors via Eigenspectrum. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 0–0.

Sitzmann, V.; Chan, E.; Tucker, R.; Snavely, N.; and Wetzstein, G. 2020. Metasdf: Meta-learning signed distance functions. *Advances in Neural Information Processing Systems*, 33: 10136–10147.

Snell, J.; Swersky, K.; and Zemel, R. 2017. Prototypical networks for few-shot learning. In *Advances in neural information processing systems*, 4077–4087.

Sun, Q.; Liu, Y.; Chua, T.-S.; and Schiele, B. 2019a. Meta-transfer learning for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 403–412.

Sun, W.; Min, X.; Zhai, G.; Gu, K.; Duan, H.; and Ma, S. 2019b. MC360IQA: A Multi-channel CNN for Blind 360-Degree Image Quality Assessment. *IEEE Journal of Selected Topics in Signal Processing*, 14(1): 64–77.

Tan, M.; and Le, Q. V. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946*.

Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.

Xiong, C.; Shukla, N.; Xiong, W.; and Zhu, S.-C. 2016. Robot learning with a spatial, temporal, and causal and-or graph. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2144–2151. IEEE.

Yang, Y.; Li, Y.; Fermuller, C.; and Aloimonos, Y. 2015. Robot learning manipulation action plans by" watching" unconstrained videos from the world wide web. In *Twenty-ninth AAAI conference on artificial intelligence*. Citeseer.

Yu, L. F.; Yeung, S. K.; Tang, C. K.; Terzopoulos, D.; Chan, T. F.; and Osher, S. J. 2011. Make it home: automatic optimization of furniture arrangement. *ACM Transactions on Graphics (TOG)-Proceedings of ACM SIGGRAPH 2011, v. 30,(4), July 2011, article no. 86*, 30(4).

Zhai, G.; and Min, X. 2020. Perceptual image quality assessment: a survey. *SCIENCE CHINA Information Sciences*, 63(11): 211301.

Ziebart, B. D.; Maas, A. L.; Bagnell, J. A.; and Dey, A. K. 2008. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, 1433–1438. Chicago, IL, USA.