One-shot Transfer Learning for Population Mapping

Erzhuo Shao1,2, Jie Feng1,2, Yingheng Wang2, Tong Xia1,2, Yong Li†1,2
1Beijing National Research Center for Information Science and Technology (BNRist)
2Department of Electronic Engineering, Tsinghua University, Beijing, China, 100084

shaoerzhuo@gmail.com, fengj12ee@hotmail.com, wangyh20@mails.tsinghua.edu.cn
xia-t17@tsinghua.org.cn, liyong07@tsinghua.edu.cn

ABSTRACT

Fine-grained population distribution data is of great importance for many applications, e.g., urban planning, traffic scheduling, epidemic modeling, and risk control. However, due to the limitations of data collection, including infrastructure density, user privacy, and business security, such fine-grained data is hard to collect and usually, only coarse-grained data is available. Thus, obtaining fine-grained population distribution from coarse-grained distribution becomes an important problem. To tackle this problem, existing methods mainly rely on sufficient fine-grained ground truth for training, which is not often available for the majority of cities. That limits the applications of these methods and brings the necessity to transfer knowledge between data-sufficient source cities to data-scarce target cities.

In knowledge transfer scenario, we employ single reference fine-grained ground truth in target city, which is easy to obtain via remote sensing or questionnaire, as the ground truth to inform the large-scale urban structure and support the knowledge transfer in target city. By this approach, we transform the fine-grained population mapping problem into a one-shot transfer learning problem.

In this paper, we propose a novel one-shot transfer learning framework PSRNet to transfer spatial-temporal knowledge across cities from three views. From the view of network structure, we build a dense connection-based population mapping network with temporal feature enhancement to capture the complicated spatial-temporal correlation between population distributions of different granularities. From the view of data, we design a generative model to synthesize fine-grained population samples with POI distribution and the single fine-grained ground truth in data-scarce target city. From the view of optimization, after combining above structure and data, we propose a pixel-level adversarial domain adaption mechanism for universal feature extraction and knowledge transfer during training with scarce ground truth for supervision.

Experiments on real-life datasets of 4 cities demonstrate that PSRNet has significant advantages over 8 state-of-the-art baselines by reducing RMSE and MAE by more than 25%. Our code and datasets are released in Github.

CCS CONCEPTS

• Computing methodologies → Neural networks; Supervised learning by regression; Artificial intelligence.

KEYWORDS

Transfer Learning, Population Distribution, Super-resolution

ACM Reference Format:

Erzhuo Shao1,2, Jie Feng1,2, Yingheng Wang2, Tong Xia1,2, Yong Li†1,2. 2021. One-shot Transfer Learning for Population Mapping. In Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM ’21), November 1–5, 2021, Virtual Event, QLD, Australia. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3459637.3482460

1 INTRODUCTION

Figure 1: Coarse-grained and fine-grained population distribution in CITY1 and CITY2. Lighter places have higher population.

Fine-grained urban population distribution, e.g., the real-time population in 100m × 100m grids in the city, is of great importance for many applications, including urban planning, epidemic modeling, and transportation management. For example, with a large-scale data recording dynamic fine-grained population distribution, governments can make timely and effective policies about infrastructure construction for commute and public health during the pandemic disease. However, due to the limitations of current data collection systems along with the issues of user privacy and business security, such fine-grained population distribution is difficult to obtain and rarely to be open. Usually, only coarse-grained population distribution like 2km × 2km is obtainable. Thus, as Figure 1, inferring fine-grained urban population distribution from coarse-grained data, also known as fine-grained population mapping problem [2, 19], becomes an important task.
Although existing neural network-based population mapping methods [14, 28] have promising performance, these methods always rely on sufficient fine-grained population distribution as ground truth to supervise their training, which severely restricts their applications since sufficient fine-grained population data are usually not obtainable. Therefore, a method, which could infer fine-grained population distribution without sufficient fine-grained ground truth, would significantly broaden the applications of population mapping in data-scarce cities. Fortunately, we could obtain prior knowledge of these cities from several approaches. First, we could extract universal and transferable knowledge about the relationship between coarse-grained population and fine-grained population from data-sufficient source city to support the population mapping in data-scarce target city. Second, although sufficient fine-grained population distribution data is unobtainable, a single static reference fine-grained ground truth is still easy-to-obtain by remote sensing [19] or questionnaire. Third, POI (Point of Interest) distribution characterizes the large-scale urban structures and region functions. Its strong relationship with population and crowd flow makes it informative.

Thus, we transform the population inference problem into a one-shot transfer learning problem, since there is only one target domain’s fine-grained ground truth in our scenario. We need to extract transferable knowledge from the data-sufficient source domain (city) and transfer it into data-scarce target domain (city) with the support of only one reference static population distribution sample and auxiliary data (i.e., POI distribution). To solve this one-shot transfer learning problem, there are still several challenges:

- **First**, the spatial-temporal correlations between the coarse and fine-grained population distribution are complicated. In the source data-sufficient domain, we could train a model with both coarse and fine-grained population data. However, the correlations between two distributions are spatially affected by the urban structure and they change in different locations. Thus, effectively learn transferable spatial-temporal correlations is challenging.

- **Second**, it’s hard to utilize the target domain’s scarce reference data to guide knowledge transfer. Single reference fine-grained population ground truth and POI distribution in the target domain could guide the knowledge transfer from the source domain. However, following experiments will prove that straightforward methods, including employing single reference fine-grained distribution as the ground truth to fine-tune model which is pre-trained in a single source domain or meta-pre-trained in multiple source domains, fail to perform well to adapt the model to the target domain. Considering the differences of large-scale structures between source and target domains, a more effective method is required for domain adaptation.

Confronting these challenges, we propose a novel Population Super-Resolution Network (PSRNet), which follows the procedure of two stages, including pre-training for knowledge extraction in the source domain and fine-tuning for knowledge transfer from source to target domain. To extract and transfer spatial-temporal knowledge, PSRNet consists of three components:

- **Spatial-Temporal Network (STNet)** for model-based knowledge transfer, used to extract the transferable spatial-temporal correlations between coarse-grained and fine-grained population distribution.
- **Population Generation Network (PGNet)** for data-based knowledge transfer, designed to transfer the relationship between POI and gridded crowd flow. It will augment the single fine-grained ground truth in target domains.
- **Pixel-level Adversarial Domain Adaptation mechanism (PADA)** as optimization-based knowledge transfer, which could mitigate the domain shift in fine-tuning stage.

In the model-based transfer network STNet, we design a dense connection-based population mapping network to extract spatial correlations from the coarse-grained population for fine-grained population mapping. Furthermore, we design a temporal module to enhance the transferability of above population mapping network by capturing the temporal correlations of spatial features and progressively merging them into different layers of STNet.

In the data-based transfer network PGNet, we design a generative adversarial-based model to learn the transferable correlations between POI distribution and gridded crowd flow from the source domain and generate fine-grained population distribution in the target domain. Concretely, we utilize the dynamic representation from time-enabled long short-term memory network (LSTM) as weights to reorganize the static urban POI map and generate the sequential gridded crowd flow with a residual convolution-based network. Then, we combine the single reference fine-grained ground truth with generated crowd flow to synthesize multiple fine-grained population distribution samples in the target domain to provide more ground truth for fine-tuning.

Finally, we enable the knowledge transfer from the optimization view. We develop a pixel-level adversarial domain adaptation framework (PADA) to adapt our model into target domains by mitigating the domain shift between different domains in the fine-tuning stage. When employing PADA for fine-tuning, except for STNet’s regular population mapping, a pixel-level discriminator is simultaneously trained to distinguish the domain of STNet’s feature maps, whereas STNet is also optimized to confuse the discriminator. With this adversarial mechanism, we could adapt STNet while ensure its feature extraction is universal for source and target domains. That would mitigate the domain shift and improve the performance of transfer.

Our contributions are summarized as follows.

- We present the first attempt in one-shot transfer learning for fine-grained population mapping, which is of great importance to deploy population mapping on data-scarce cities. Concretely, we develop a novel framework with three-view knowledge transfer mechanisms to infer fine-grained population distribution with scarce data in the target domain.

- We design a model-based transfer network STNet to transfer the spatial-temporal correlations between coarse-grained and fine-grained population distribution by its parameters. Besides, we develop a data-based transfer model PGNet to synthesize fine-grained ground truth in target domains and transfer the correlation between POI distribution and gridded crowd flow. Finally, based on the aforementioned components, we design an pixel-level adversarial domain adaption fine-tuning framework PADA to reduce the domain shift in spatial-temporal knowledge.
transfer between source and target domains during fine-tuning optimization.  

- We conduct extensive experiments on real-life datasets of 4 cities to evaluate the performance of our proposed model, including the knowledge transfer across cities and granularities. Results of on four metrics in x2 and x4 tasks demonstrate that our model has significant advantages over 8 state-of-the-art baselines.

2 PRELIMINARIES

In this section, we first introduce the notations and then formally define the fine-grained population mapping problem in the one-shot transfer learning scenario. Following previous works [14, 19, 28], we use gridded population distribution to formulate the fine-grained population mapping problem.

Definition 1 (Gridded Population Distribution). By partitioning an area into a $H \times W$ grid map, the gridded population distribution in a single time slot is defined as tensor $X \in \mathbb{R}^{T \times H \times W}$ by accumulating the users visiting each grid. The sequential population distribution with consecutive $T$ time slots in source domain and target domain are denoted as $X^s \in \mathbb{R}^{T \times H \times W}$, $X^t \in \mathbb{R}^{T \times H \times W}$.

Under above settings, population could be mapped into grid maps of different granularities (e.g., 500m x 500m or 2km x 2km). Coarse-grained population distribution, with grid size 2km x 2km is denoted by $X^c \in \mathbb{R}^{T \times H \times W}$. Fine-grained population distribution e.g., with grid size 500m x 500m is denoted by $X^f \in \mathbb{R}^{T \times H \times W}$, $(n = 2000/500 = 4)$. We note that both coarse-grained and fine-grained population are relative and task-specific, which will be introduced before each comparison in Experiments 4. In this research, 1 time slot always contains 30 minutes. The fine-grained population mapping task needs to recover the fine-grained distribution from coarse-grained distribution, which is formally defined as below:

Problem 1 (Fine-grained Population Mapping). Given the coarse-grained population distribution sequence $X^c \in \mathbb{R}^{T \times H \times W}$ of $T$ time slots (e.g., from 06:00PM to 09:00PM), estimate the fine-grained population distribution of the newest ($T$th) time slot $X^f \in \mathbb{R}^{T \times H \times W}$ (e.g., at 09:00PM). We note that $n$ is the upscale factor, which means population in each grid need to be partitioned into $n \times n$ sub-grids.

While fine-grained population data is difficult to obtain for the majority of cities in practice, we attempt to transfer knowledge from data-sufficient city to data-scarce city with the support of single reference static fine-grained population and POI distribution in target city. The single reference static fine-grained population is available via remote sensing [19] or questionnaire. It is critical to indicate large-scale urban structure and patterns of population in the data-scarce target domain, which is denoted by $X^f_{ref} \in \mathbb{R}^{T \times H \times W}$. Moreover, POIs distribution characteristic features of population. It is also considered as a reliable and informative proxy of human activity [4, 16, 24, 26] in target domain. With partitioning an area into a grid map, the number of POIs of each category is defined as a tensor $P \in \mathbb{R}^{C \times H \times W}$ by accumulating POIs in each grid into $C$ categories.

By introducing knowledge transfer, single fine-grained population distribution and POI distribution for target city, we transform the fine-grained population mapping task into a one-shot transfer learning problem, since there is only one fine-grained population distribution as ground truth in target domain. This problem is formally defined as follows:

Problem 2 (One-shot Transfer Learning For Fine-grained Population Mapping). Given:

- Sufficient coarse and fine-grained population distribution $X^c \in \mathbb{R}^{L \times T \times H \times W}$, $X^f \in \mathbb{R}^{L \times T \times H \times W}$ of source domain $S$, where $L >> 1$ is the number of samples.
- Sufficient coarse-grained population distribution $X^c \in \mathbb{R}^{L \times T \times H \times W}$, one static reference fine-grained population distribution sample $X^f_{T,ref} \in \mathbb{R}^{T \times H \times W}$ in target domain $T$.
- Fine-grained POI distributions (static) $P^s \in \mathbb{R}^{C_{s} \times H \times W}$, $P^t \in \mathbb{R}^{C_{t} \times H \times W}$ in source and target domains.

Estimate the fine-grained population distribution $X^f \in \mathbb{R}^{L \times T \times H \times W}$ in target domain $T$.

3 METHODS

To solve the fine-grained population mapping problem in the one-shot transfer learning scenario, we propose a novel model PSRNet, whose basic procedure is presented in Figure 2. PSRNet consists of three components: STNet for model-based knowledge transfer, PGNet for data-based transfer, and pixel-level adversarial domain adaptation (PADA) for optimization-based knowledge transfer. Firstly, Our STNet is designed to complete the population mapping task by modeling the complicated spatial-temporal correlations between coarse and fine-grained population distributions. Further, it is enhanced by temporal modeling network TNet. Secondly, PGNet is designed to generate gridded crowd flow and synthesize fine-grained population distribution ground truth by capturing the spatial-temporal correlations between gridded crowd flow and POI distribution via a generative adversarial network (GAN). Finally, with the combination of STNet and PGNet, we propose PADA for optimization-based knowledge transfer, which encourages the model to transfer the spatial-temporal knowledge while mitigating the domain shift between different domains. In this way, our model PSRNet succeeds in one-shot transfer learning for the fine-grained population mapping problem.

The training procedure of PSRNet is described as follows:

- Pre-training: For STNet, We employ sufficient data in source domain to infer fine-grained population distribution by sequential coarse-grained population distribution. We also train PGNet to synthesize fine-grained gridded crowd flow by POI distribution in the source domain.
3.1 STNet: Spatial-Temporal Correlation Modeling

**Fine-tuning:** First, we employ PGNet, which is pre-trained in source domain, to generate gridded crowd flow with POI distribution in target domain. Second, we combine the single reference fine-grained population distribution and the generated gridded crowd flow to synthesize fine-grained population distribution in target domain. Third, we employ the synthetic fine-grained population distribution as ground truth to support the PADA mechanism to adapt the STNet into target domain.

### 3.1.1 SNet for Spatial Modeling

STNet is designed to extract universal spatial-temporal correlations from the coarse-grained population input \(X^c \in \mathbb{R}_+^{T \times H \times W}\) and produce the fine-grained distribution \(X^f \in \mathbb{R}_+^{L \times H \times W}\). As Figure 3 shows, it consists of two components: the first part is the backbone network SNet, which is for the spatial modeling of single input of coarse-grained population \(X^c \in \mathbb{R}_+^{T \times H \times W}\) in the Tth time slot; the second part is the temporal enhancement network TNet, which is designed to enhance SNet by modeling the temporal correlation of the sequential coarse-grained population input \(X^c \in \mathbb{R}_+^{T \times H \times W}\). Now, we discuss the details of these two networks.

#### 3.1.1.1 SNet for Spatial Modeling

We first introduce the backbone network SNet for spatial modeling. As Figure 3 shows, SNet can be divided into three parts: 1) the feature extraction unit for input preprocessing and preliminary feature extraction from the coarse-grained population; 2) stacked conv-blocks for advanced spatial feature extraction; and 3) the upsampling components to produce the fine-grained population map based on the feature map from previous feature extractors. We design two types of feature extractors. As shown at the bottom of Figure 3, the preliminary feature extraction unit is made up of two 5 × 5 convolution units which are activated by a ReLU function. Here, we choose 5 × 5 filter to expand the receptive field of feature pre-processing. Following the preliminary feature extraction unit, we stack several dense connected conv-blocks as the advanced feature extractor to extract and fuse features again. The detailed design of conv-block unit is presented at the bottom of Figure 3. It consists of a two-layer 3 × 3 convolution unit activated by ReLU function and a batch-norm layer after the first 3 × 3 convolution layer. Based on this basic unit, we apply the dense connection to construct the conv-block.

After merging all the output features from two levels of feature extractors with a 1 × 1 convolution layer, we build an up-sampling unit to upscale the feature map. Each up-sampling unit is designed to upscale the feature map by 2 times. One up-sampling unit consists of three layers: a 3 × 3 convolution layer with batch-norm layer, a pixel-shuffle layer [17] with scale 2 for rearranging and up-scaling, and a ReLU function for non-linear activation. Stacked several up-sampling units or pixel-shuffle layer of higher scale could achieve a larger up-scaling size. Different from the general image super-resolution task, the fine-grained population mapping task exhibits a specific value constraint: the population of an area equals to the total population of its sub-areas. Therefore, we finally follow \(N^2\)-Normalizaiton [14] to achieve refine the fine-grained population.

### 3.1.2 TNet for Temporal Enhancement

While SNet is designed for single input of coarse-grained population \(X^c \in \mathbb{R}_+^{T \times H \times W}\), a simple extension method of it for the temporal modeling is to process the sequential input by directly concatenating \(T\) consecutive time slots in the channel dimension as the sequential input \(X^c \in \mathbb{R}_+^{T^2 \times H \times W}\). However, simple concatenation in the channel dimension is limited to capture this long-term correlation, which is testified in our Ablation Study 4. Due to the regularity and daily periodicity of dynamic population distribution, we need to consider long-term effects in the temporal modeling. Thus, we design TNet to capture this important long-term temporal correlation, which is shown on the top of Figure 3.

To model these long-term effects and avoid the limitation of simple concatenation in channel dimension, we first utilize a 3D convolution layer with stride \(T_S\) to merge the coarse-grained population input in adjacent time slots. The parameter \(T_S\) of stride step denotes the time window of merging features as an important factor for trading performance and model complexity. Then, we utilize the feature extraction unit with the same structure in SNet to process each merged feature independently. For example, for \(T_S = 6\) and \(T = 48\), the number of the merged features is \(L = 48/6 = 8\). We use this shared feature extraction unit to process these merged features and produce new features with the same number. Finally, we progressively merge these \(L\) features into \(L\) conv-blocks. For example, the first feature map is fed into the first conv-block and the second map feature is fed into the second conv-block. In this way, the temporal features of \(L\) different periods are merged into \(L\) different layers of the spatial modeling network SNet, which can ensure that each merging operation only needs to process fewer features. It can also be regarded as an interpretative weighting scheme, more important temporal features are fed into the earlier location of the whole structure.

In summary, we design a dense connection-based SNet to complete fine-grained population mapping task from the spatial view and then design TNet to enhance the SNet from the temporal view to further improve the results.
3.2 PGNNet: Static Fine-grained Population Distribution Synthesis

3.2.1 Overview. Facing the problem of lacking fine-grained population ground truth in target domain, we propose a generative model, PGNNet, to generate gridded crowd flow by POI distribution. Combined with the single reference fine-grained population, PGNNet could synthesize multiple fine-grained population distribution.

Population distribution, crowd flow, and POI distribution are highly associated so it’s reasonable to infer one of them based on others. Firstly, the static reference fine-grained population distribution is strongly associated with crowd flow or population. Thus it’s strongly associated with crowd flow or population.

Given the close relationship between POI, population, and crowd flow, our PGNNet learns transferable universal knowledge of the correlations between POI distribution and crowd flow in the source domain and generates fine-grained crowd flow in target domain. Then, we synthesize the fine-grained population of more time slots by accumulating the generated crowd flow onto single fine-grained reference population distribution. These synthetic population distribution data contain the knowledge of the target domain’s static reference distribution, target domain’s POI distribution. Finally, these synthetic population distributions will be employed as the ground truth in fine-tuning. The whole generative framework of PGNNet is presented in Figure 4, which contains a generator to produce dynamic fine-grained population distribution from time-weighted POI density and a discriminator to produce learning signals for the generator via distinguishing whether the input distribution is synthetic or real.

3.2.2 Detailed Designs. We first introduce the design of the generator, which is at the left of Figure 4. As the goal of human activities to move to different regions, POI describes urban function and urban structure to a large extent [26]. Here, we try to generate the fine-grained crowd flow from dynamic weighted fine-grained POI density map. First, to generate the fine-grained crowd flow in time $t$ of the day, we use a learnable time embedding table to convert time $t$ into a dense feature vector. Then we feed this time vectors into a LSTM to produce the sequential representation $T_t \in \mathbb{R}^{G \times 1 \times 1}$ of time $t$. On the other hand, given the fine-grained POI distribution $P \in \mathbb{R}^{C \times H \times W}$, we utilize a 5x5 convolution layer and 2 res-blocks to pre-process it and obtain its feature map $P' \in \mathbb{R}^{G \times H \times W}$. Then, we multiply time embedding onto each pixel of $P'$ to obtain a time-aware feature map $P''$. Moreover, we stack 4 res-blocs and a 5 x 5 convolution layer to further process it to generate the fine-grained crowd flow map $\Delta \hat{X'}[t] \in \mathbb{R}^{N \times H \times W}$ of time $t$. Finally, we generate the population distributions of more time slots by adding generated crowd flow $\Delta \hat{X'}[t]$ onto static reference fine-grained distribution $X_{ref}$ and employ a $N^2$-Normalization layer to regularize these population maps. The whole generator is formulated as follows:

$$\Delta \hat{X'}[t] = X'[t+1] - X'[t].$$  
(1)

$$\Delta \hat{X'}[t] = \mathcal{F}_2(\text{Reshape}(\text{LSTM}([\mathcal{F}_1(t)])) + \mathcal{F}_1(P)), $$  
(2)

$$\hat{X'}[1+1] = \text{Norm}(X'_{ref} + \Sigma_{l=0}^{N} \Delta \hat{X'}[t]),$$  
(3)

$$\hat{X'}[-l-1] = \text{Norm}(X'_{ref} - \Sigma_{l=0}^{N} \Delta \hat{X'}[t])$$  
(4)

We first define fine-grained gridded crowd flow $\Delta \hat{X'}[t]$ as the difference fine-grained population distribution between two consecutive time slots $[t+1]$ and $[t]$. $\mathcal{F}_2$ represents the embedding layer, $\text{Reshape}$ denotes the vector extension operation, $\mathcal{F}_1$ and $\mathcal{F}_2$ denotes the two feature extraction module including convolution layers and res-blocks, $P \in \mathbb{R}^{C \times H \times W}$ denotes the category-aware POI distribution, $\text{Norm}$ stands for the $N^2$-Normalization layer, $\Delta \hat{X'}[t]$ means the estimated fine-grained crowd flow in time $t$, static reference fine-grained population distribution is $X'_{ref}$, $\hat{X'}[t+1]$ and $\hat{X'}[-l-1]$ represent the synthetic fine-grained population distribution in forward and backward directions. The sequence of fine-grained population distribution $[\hat{X'}[-l-1] \cdots \hat{X'}[t+1]] = \hat{X'} \in \mathbb{R}^{C \times H \times W}$ is the final output of PGNNet’s generator, where $F$ is the length of output sequence.

To guide the learning of the generator, we build a discriminator to generate the learning signals. The discriminator consists of two major components: the convolution-based feature extractor which consists of a $3 \times 3$ convolution unit and 3 res-blocks and a classification module with a linear layer activated by sigmoid function. The training of PGNNet follows the standard GAN training procedure with a combined loss of GAN loss and MSE loss with weighting coefficient $\alpha$ as a hyper-parameter.

![Figure 4: The framework of PGNNet in PSRNet.](image-url)
3.3 Pixel-level Adversarial Domain Adaptation

Cooperating with the model-based transfer by STNet and data-based transfer by PGNet, we introduce the optimization-based transfer by adversarial domain adaptation training mechanisms, which is presented in Figure 2. Adversarial domain adaptation [6, 20] is an effective transfer learning algorithm. We adopt its basic structure while adapting it into our problem, which contains three components:

- **TNet - Feature Extractor**: We first utilize the TNet as a universal Feature Extractor to extract feature maps in source and target domains. It is optimized to support the regression of Predictor. Whilst, it is also optimized against the Domain Classifier to confuse its classification task.

- **SNet - Predictor**: With pre-trained PGNet, we obtain the synthetic fine-grained population data \( X^f_T \) in target city \( T \) as ground truth. With the synthetic multiple fine-grained population distributions as ground truth, Predictor and Feature Extractor are optimized to complete the prediction task with MSE loss.

- **Domain Classifier**: Domain Classifier accepts the concatenation of feature maps from source and target domains. The classifier is highly similar to PGNet but we remove the last 3 layers and employ multi-layer perceptron (MLP) to directly classify the domain of each pixel, which contains multiple channels. Domain Classifier is optimized to classify the domain of input feature maps by working with Predictor parallely.

We repeat the optimization until convergence. Finally, we get a well-trained STNet to complete the fine-grained population mapping task on the target domain by learning the universal spatial-temporal knowledge between cities from the model-based, data-based, and optimization-based transfer views.

4 EXPERIMENTS

4.1 Dataset

We employ real-life datasets from 4 cities, which are represented by CITY1 - CITY4, to evaluate the performance of models. These datasets are collected from mobile devices by a popular mobile localization service provider in China, which is dense in the population level and thus close to the real population distribution. It covers 4 cities with a duration of 1 month (2018.08~2018.09). It records the locations whenever users request localization services in the applications. To obtain the fine-grained gridded population distribution, each location record is projected into a grid in 500m \( \times \) 500m chessboard as the finest granularity, while timestamp is projected into time windows of 30 minutes. Records are aggregated by counting the population value of each grid in each time window. We noted that the raw data with anonymous individual information is not available and we could only access the aggregated population data.

We also collect POI data in these cities from the public website of the localization service. For each city, we collect about 1 million POI instances and calculate the category-based POI distribution map, which would be used in PSRN. These POIs are classified into 14 categories, including food, hotel, culture, sports, shopping, factory, recreation, institution, medical care, scenic spot, education, landmark, residence, travel & transport, business affairs, and life service.

4.2 Baselines

To evaluate the performance of our model, we compare it with 8 state-of-the-art baselines, including 2 traditional methods (Bicubic and LightGBM), 2 super-resolution based methods for fine-grained population mapping task (DeepDPM and UrbanFM), and 4 advanced methods for image and video super-resolution (RCAN, DBPN, RRN, and RBPN).

- **Bicubic** [7]: A widely used up-sampling method for image processing, we use it to up-sample the coarse-grained population distribution.

- **LightGBM** [11]: It is a gradient boost regression tree-based ensemble semi-automated machine learning method, which is highly efficient and effective.

- **DeepDPM** [28]: It contains a CNN-based spatial mapping network and an RNN-based temporal smoothness network to extract spatial-temporal features and achieves promising results for population mapping.

- **UrbanFM** [14]: With its ResNet-based super-resolution network and \( N^2 \)-Normalization layer, UrbanFM achieves state-of-the-art performance on the fine-grained urban flow inference task.

- **RCAN** [27]: It improves its performance of image super-resolution by considering the residual connection and channel attention in the model.

- **DBPN** [8]: With back-projection units and dense connection, it repetitively up-samples and down-samples the feature maps and concatenates them for high-resolution image reconstruction.

- **RRN** [10]: By processing the feature map with its residual module recurrently, RRN is capable to complete the video super-resolution task.

- **RBPN** [9]: As the state-of-the-art method for video super-resolution, it is an enhanced version of DBPN by considering the temporal correlation with a multiple projection mechanism.

4.3 Experimental Settings

In the pre-processing, \( n \times n \) task means with a sequence of fine-grained population distribution of shape \( T \times nH \times nW \) in a certain dataset, we add the population of adjacent \( n \times n \) grids together and get a sequence of coarse-grained population maps with shape \( T \times nH \times nW \). Then we use \( T \times nH \times nW \) distribution to infer \( T \times nH \times nW \) distribution. Fine-grained POI distribution and single reference fine-grained population distribution are always of fine-grained shape \( C \times nH \times nW \) and \( 1 \times nH \times nW \).

For source cities, we randomly select 70% time slots as the training set and utilize the remaining 15% and 15% time slots as validation set and test set. For target cities, we use 1 week as the test set to evaluate the performance on the fine-grained population mapping task in the transfer learning scenario, while its first time slots is used as the reference fine-grained population distribution in target domain, it can be regarded as a one-shot transfer learning scenario. The default settings of STNet are length of population sequence \( T = 48 \), time stride \( T_S = 6 \), number of layers \( L = 48/6 = 8 \), base
input/output channel \( C_B = 64 \), time channel \( C_T = 16 \). For the generator of \( PGNet \), we use base channel \( C_G = 64 \). For the discriminator of \( PGNet \), we use number of layers \( L = 3 \), spatial stride \( S_S = 4 \), base channel \( C_D = 1 \).

We evaluate the model with 4 metrics: Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Pearson Correlation Coefficient (Corr). Based on these metrics, we calculate the error between estimated fine-grained population distributions and their ground truth.

We conducted experiments on Ubuntu 18.04.3 LTS system with 4x NVIDIA GTX 2080Ti using Python 3.6.10 and PyTorch 1.6.0. Our models, experiment code, and datasets are available via Github.

### 4.4 One-shot Transfer across Cities

To verify the effectiveness of our \( PSRNet \), we compare our model with baselines on the \( \times 2 \) and \( \times 4 \) fine-grained population mapping task in the cross-cities scenarios.

For all baselines, we firstly pre-train their models in CITY1 with sufficient data and employ MAML [5] on all cities except the target city as an additional meta-pre-training. Then we use the single reference fine-grained population distribution to fine-tune them on the target city (CITY2 or CITY3) and obtain the final mapping results.

In this subsection, to train our proposed \( PSRNet \), firstly, we employ CITY1 to pre-train \( STNet \) and \( PGNet \). Secondly, we employ the pre-trained \( PGNet \) to synthesize distribution in more time slots with the single reference fine-grained population distribution. Finally, we employ \( PADA \) 3.3 to fine-tune the pre-trained \( STNet \) with the synthetic population distribution. Then, \( PSRNet \) is enabled to generate fine-grained population distribution. We note that our \( PSRNet \) is not only compared with the architecture of baselines but also compared with MAML [5], which is the meta-learning mechanism for knowledge transfer across cities. We note that \( PSRNet \) only utilize 1 source city (CITY1) for pre-training while 3 source cities are used to support baselines’ meta-learning. That would bring advantages for baselines in this comparison.

Table 1 shows the performance of all baselines and our \( PSRNet \), where the notation Improv. indicates the percentage of reduction of RMSE, MAE, MAPE, and the increase of Corr of our method when compared with the best baselines. From these results, we can find that \( PSRNet \) always outperforms all baselines in all metrics. Although \( UrbanFM \), \( DBPN \), \( RRN \), and \( RBPN \) reach comparable results in some scenarios, none of them could outperform \( PSRNet \) in any task. They obtain 16, 546, 19, 8083, 20, 8997, and 27, 7215 by RMSE, which are the second-best results in \( \times 2 \) and \( \times 4 \) tasks when CITY2 and CITY3 are target cities. Compared with these results, \( PSRNet \) reduces the RMSE by 14.4%, 18.3%, 19.9%, and 24.7%, while other baselines could only achieve comparable performance in minor metrics or tasks, and fail to sustain a consistently comparable performance in other tasks. \( UrbanFM \) is the only comparable baseline since is specially designed for urban population mapping scenario, while it is still less competitive than \( PSRNet \).

| Dataset | CITY2 (\( \times 2 \)) | CITY3 (\( \times 2 \)) | CITY2 (\( \times 4 \)) | CITY3 (\( \times 4 \)) |
|---------|----------------------|----------------------|----------------------|----------------------|
| Metrics | RMSE | MAE | MAPE | Corr | RMSE | MAE | MAPE | Corr | RMSE | MAE | MAPE | Corr | RMSE | MAE | MAPE | Corr |
| Bibubic | 19.46 | 12.97 | 0.32 | 0.87 | 37.54 | 20.45 | 0.45 | 0.85 | 25.78 | 17.18 | 0.41 | 0.75 | 50.48 | 28.98 | 0.64 | 0.73 |
| LightGBM [11] | 19.36 | 12.75 | 0.30 | 0.85 | 37.43 | 20.01 | 0.44 | 0.65 | 28.19 | 17.64 | 0.40 | 0.76 | 50.60 | 28.64 | 0.64 | 0.70 |
| DeepFM [18] | 19.90 | 13.57 | 0.34 | 0.87 | 32.13 | 17.21 | 0.38 | 0.79 | 25.73 | 16.92 | 0.42 | 0.79 | 41.91 | 21.18 | 0.47 | 0.84 |
| UrbanFM [14] | 16.54 | 10.64 | 0.26 | 0.91 | 19.19 | 10.59 | 0.23 | 0.95 | 20.90 | 13.49 | 0.33 | 0.86 | 27.72 | 13.65 | 0.30 | 0.92 |
| RCAN [27] | 17.38 | 18.80 | 0.25 | 0.94 | 33.60 | 19.25 | 0.41 | 0.92 | 21.18 | 14.55 | 0.36 | 0.85 | 28.72 | 16.57 | 0.37 | 0.92 |
| DBPN [8] | 17.74 | 11.40 | 0.28 | 0.90 | 20.48 | 11.35 | 0.25 | 0.96 | 25.23 | 16.12 | 0.40 | 0.81 | 29.92 | 16.95 | 0.37 | 0.91 |
| RRN [10] | 18.73 | 11.50 | 0.28 | 0.90 | 34.07 | 19.56 | 0.43 | 0.90 | 38.97 | 25.18 | 0.61 | 0.69 | 50.08 | 31.83 | 0.72 | 0.79 |
| RBPN [9] | 17.90 | 12.14 | 0.30 | 0.91 | 23.39 | 14.58 | 0.32 | 0.94 | 22.85 | 15.61 | 0.39 | 0.85 | 31.62 | 19.19 | 0.42 | 0.90 |
| OurBest | 14.15 | 8.39 | 0.21 | 0.94 | 16.17 | 8.247 | 0.18 | 0.97 | 16.17 | 8.247 | 0.18 | 0.97 | 20.86 | 10.28 | 0.20 | 0.95 |
| Improv. | 14.4% | 21.1% | 21.1% | 7.4% | 18.3% | 22.2% | 22.2% | 8.0% | 19.9% | 24.3% | 24.3% | 8.0% | 24.7% | 24.7% | 24.7% | 19.3% |

In addition to the numerical evaluation of models’ performance, we also compare them by visualization. The ground truth map and predicted fine-grained population maps at 09:00 AM of PSRNet, UrbanFM, and RBPN in CITY2’s \( \times 4 \) task are demonstrated in Figure 5.

In these figures, the lighter places denote greater population values and vice versa. The region circled by the smaller white rectangle is a region that contains 8 \( \times 8 \) fine-grained grids mapped from 2 \( \times 2 \) coarse-grained grids in \( \times 4 \) task. The bigger rectangles are 3X zooms of smaller rectangles. We note that the river crosses the city from the bottom to the right-up corner of the image and the selected region is on the bank of this river.

In Figure 5, PSRNet’s distribution have the closest shapes and textures with the ground truth distribution. For example, compared with PSRNet, UrbanFMs’ predicted fine-grained population suddenly changes on the margins of different coarse-grained pixels so the zoomed area has an obvious unnatural vertical dividing line. Besides, the map of RBPN is much lighter and rougher than PSRNet in the zoomed region. These results show that compared with RBPN, PSRNet could capture the pattern of the river and produce more reasonable fine-grained distributions.

In summary, numerical comparison and visualization prove that the proposed PSRNet has significant advantages over state-of-the-art baselines in fine-grained population mapping tasks in the cross-cities transfer learning scenario.

### 4.5 One-shot Transfer across Granularities

In this subsection, we research an extreme data-scarce scenario in which any data out of the target cities is unobtainable. In this
Table 2: Performance of our model and baselines on cross-granularities knowledge transfer task. (2km→1km)→(1km→500m) means we pre-train PSRNet in source domain, in which we infer population of 1km × 1km from population of 2km × 2km, while in target domain, we infer 500m × 500m from 1km × 1km.

| Dataset | CITY2          | CITY3          |
|---------|----------------|----------------|
|         | (2km→1km)→(1km→500m) | (2km→1km)→(1km→500m) |
| Metrics | RMSE MAE MAPE Corr | RMSE MAE MAPE Corr |
| Bibubic | 19.463 12.973 0.325 0.874 | 37.904 20.454 0.457 0.850 |
| LightGBM| 22.205 14.033 0.352 0.852 | 41.810 21.799 0.488 0.839 |
| DeepDPFM| 19.698 13.314 0.334 0.873 | 34.333 18.854 0.422 0.879 |
| UrbanFM | 27.677 11.219 0.281 0.906 | 21.222 11.647 0.260 0.953 |
| RCAN   | 19.795 12.995 0.318 0.859 | 30.956 19.144 0.428 0.900 |
| DBPN   | 18.402 12.102 0.303 0.898 | 24.980 14.237 0.318 0.936 |
| RRN    | 19.195 13.181 0.330 0.888 | 31.045 17.803 0.398 0.902 |
| RDBN   | 21.382 14.458 0.362 0.859 | 35.185 21.869 0.489 0.881 |
| PSRNet | 13.973 7.590 0.199 0.950 | 16.164 7.886 0.176 0.972 |
| Improv. | 26.0% 29.1% 29.1% 4.8% | 23.8% 32.3% 32.3% 2.8% |

4.6 Ablation Study

In this section, we conduct an ablation study in cross-cities scenario to evaluate the effectiveness of each proposed component of PSRNet. To investigate their effect, we compare different variants of our method. To evaluate SNet’s capability of spatial-temporal knowledge extraction, UrbanFM is also introduced in this comparison. All the variants are introduced as follows:

- **SNet** is the backbone population mapping network of PSRNet, which accepts 48 coarse-grained population maps as multiple input channels.
- **TNet** denotes that we introduce TNet to enhance the temporal modeling of sequential input. SNet+TNet is simplified as STNet.
- **PGNet** represents we employ PGNet to generate synthetic fine-grained population map in target domain. Then these generated maps are used to fine-tune the pre-trained population mapping network.
- **MAML** means we employ meta-learning algorithm MAML [5] to drive an additional meta-training with three cities’ data except the target city for population mapping models.
- **PADA** means employ pixel-level adversarial domain adaptation 3 to fine-tune the model in target domain. The complete version of our proposed method is STNet+PGNet+PADA (denoted as PSRNet).

The results in Table 3 brings us several conclusions:

- **Our SNet could outperform UrbanFM, which shows that SNet could capture spatial patterns in population maps by its more advanced architecture.**
- **In STNet, combined with TNet, which could effectively exploit temporal information, the feature maps of different time slots are fed into different layers. Therefore, each layer of STNet only needs to process less information. STNet outperforms SNet indicates that STNet is better to capture transferable features in all scenarios.**
- **By comparing the performance of STNet and MAML+STNet, we find that although MAML slightly improves STNet’s performance in ×2 tasks, it fails to sustain this improvement in ×4 tasks. That shows the meta-learning method failed to transfer knowledge across cities stably.**
- **SNet+PGNet outperforms SNet, while STNet+PGNet outperforms STNet in all scenarios. That shows our PGNet could always achieve the best performance in both cities and all metrics, while UrbanFM always reaches the second-best results. We note that in cross-cities scenario of ×2 task, PSRNet’s RMSE is 14.1569 and 16.1735 in CITY2 and CITY3, whereas these numbers are 13.0725 and 15.9907 in cross-granularities scenario, which shows the domain shift in the cross-granularities scenario is smaller than cross-cities scenario. Therefore, the knowledge of same city’s coarse-grained distribution is more transferable than other cities’ fine-grained distribution, which validates the aforementioned assumption. Furthermore, it also strongly implies PSRNet is potential to generate the finer-grained population distribution (i.e., of 250m × 250m or even finer granularity) as long as we employ a finer-grained static reference distribution to fine-tune PSRNet. Unfortunately, we are not able to validate the results without enough ground truth in finer granularity.**

In summary, our proposed model-based, data-based, and optimization-based transfer learning components in PSRNet bring performance gain in one-shot transfer learning fine-grained population mapping task. That proves the reasonability of our design.
5 RELATED WORKS

5.1 Image and Video Super-Resolution

With the application of deep learning, research community [3, 8, 13, 22, 27] makes significant progress on image super-resolution task. SRCNN [3] utilizes several convolution layers to build the first end-to-end framework for image super-resolution. With the basic idea of first building deep convolution networks to extract features and then upsampling to obtain the high-resolution image, many following ups works are proposed with more advanced network design [8, 27], specific loss function [12] and so on. Considering the temporal correlation between multiple frames, image super-resolution upgrades to the video super-resolution task. While some works [15] try to model spatial-temporal correlation via the motion compensation between different frames by optical flow or learning, others [9, 13] try to directly learn the spatial-temporal dependency with different sequential structures like recurrent neural networks. Different from these existing works on the general image/video super-resolution task, we focus on the fine-grained population mapping task and propose effective methods to transfer the spatial-temporal knowledge to promote the performance in cities without fine-grained data.

5.2 Fine-Grained Population Mapping

By applying the successful practice of image super-resolution into fine-grained population mapping, DeepDPM [28] and UrbanFM [14] are the most related work to our work. DeepDPM [28] first utilizes SRCNN [3] with stacking structure to up-sample the static population distribution and then utilizes LSTM to refine the population variation in the temporal dimension. To infer the fine-grained crowd flow, UrbanFM [14] proposes a ResNet-based network structure with applying the recent practice from image super-resolution and also consider the effects of external factors like holidays in the model. While they achieve promising performance in the city with enough data, they require a large number of fine-grained data to train the whole model, which is not available for most of the cities. Different from them, our work considers the transferred fine-grained population mapping task and proposes to transfer the spatial-temporal knowledge from the data and model view to improve the mapping performance on these cities without fine-grained data.

5.3 Transfer Learning Among Cities

Knowledge transferring between cities is an important topic in urban computing. Wei et al. [23] propose FLORAL with learning semantically related dictionaries and transferring dictionaries and instances to predict air quality in different cities. Wang et al. [21] propose to use slide information from public check-ins to align regions from different cities to enable the explicit knowledge transfer in the crowd flow prediction task. Yao et al. [25] apply MAML optimization methods to enable the multi-cities crowd flow prediction. These existing works focus on the multi-variant time series prediction problem and are not directly available for our mapping task. Furthermore, different from these works which only transfer knowledge from single view, our framework enables the spatial-temporal knowledge transfer from model, data, and optimization views.

6 CONCLUSION

In this paper, we investigated the fine-grained population mapping problem in the transfer learning scenario. We transfer this problem into a one-shot transfer learning problem for the population mapping task. We proposed a novel model by transferring the spatial-temporal knowledge from model view, data view, and optimization view. We designed a sequential population mapping network to capture the complicated correlation between the population of different granularities. Furthermore, we proposed a generative model to synthesize multiple fine-grained population samples in target domain with POI distribution. Finally, we utilized the adversarial adaptation methods to fine-tune the pre-trained model and transfer the universal spatial-temporal knowledge.

ACKNOWLEDGMENTS

This work was supported in part by The National Key Research and Development Program of China under grant 2018YFB1800804, the National Nature Science Foundation of China under U1936217, 61971267, 61972223, 61941117, 61861136003, Beijing Natural Science Foundation under L182038, Beijing National Research Center for Information Science and Technology under 20031887521, and research fund of Tsinghua University - Tencent Joint Laboratory for Internet Innovation Technology.
REFERENCES

[1] 2020. Points of Interest in OpenStreetMap. https://wiki.openstreetmap.org/wiki/Points_of_interest. Accessed: 2020-01-01.

[2] Pierre Delvée, Catherine Linard, Samuel Martin, Marúsia Gilbert, Forrest R Stevens, Andrea E Gauaghan, Vincent D Blondel, and Andrew J Tatem. 2014. Dynamic population mapping using mobile phone data. *Proceedings of the National Academy of Sciences* 111, 45 (2014), 15888–15893.

[3] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. 2014. Learning a deep convolutional network for image super-resolution. In *European conference on computer vision*. Springer, 184–199.

[4] Lei Dong, Carlo Ratti, and Siqi Zheng. 2019. Predicting neighborhoods’ socio-economic attributes using restaurant data. *Proceedings of the National Academy of Sciences* 116 (2019), 15447 – 15452.

[5] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*. PMLR, 1126–1135.

[6] Yaroslav Ganin and Victor S. Lempitsky. 2015. Unsupervised Domain Adaptation by Backpropagation. *ArXiv* abs/1409.7495 (2015).

[7] Rafael C. González and Richard E. Woods. 1981. Digital Image Processing. *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-3 (1981), 242–243.

[8] Muhammad Haris, Greg Shakhnarovich, and Norimichi Ukita. 2018. Deep Back-Projection Networks for Super-Resolution. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[9] Muhammad Haris, Greg Shakhnarovich, and Norimichi Ukita. 2019. Recurrent Back-Projection Network for Video Super-Resolution. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[10] Takashi Isobe, Fang Zhu, and Shengjin Wang. 2020. Revisiting Temporal Modeling for Video Super-resolution. *arXiv preprint arXiv:2008.05763* (2020).

[11] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems* 30 (2017), 3146–3154.

[12] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. 2017. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4681–4690.

[13] Sheng Li, Fengxiang He, Bo Du, Lefei Zhang, Yonghao Xu, and Dacheng Tao. 2019. Fast spatio-temporal residual network for video super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 10522–10531.

[14] Yuxuan Liang, Kun Ouyang, Lin Jing, Sijie Ruan, Ye Liu, Junbo Zhang, David S Rosenblum, and Yu Zheng. 2019. UrbanFM: Inferring Fine-Grained Urban Flows. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 3132–3142.

[15] Renjie Liao, Xin Tao, Ruiyu Li, Ziyang Ma, and Jiaya Jia. 2015. Video super-resolution via deep draft-ensemble learning. In *Proceedings of the IEEE International Conference on Computer Vision*. 533–539.

[16] Erzhuo Shao, Huandong Wang, Jie Feng, Tong Xia, Hedong Yang, Lu Geng, De-renjie Liao, Xin Tao, Ruiyu Li, Ziyang Ma, and Jiaya Jia. 2015. Video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4681–4690.

[17] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. 2017. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4681–4690.

[18] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. 2017. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4681–4690.

[19] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. 2017. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4681–4690.

[20] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. 2017. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4681–4690.

[21] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. 2017. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4681–4690.

[22] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. 2017. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4681–4690.

[23] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. 2017. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4681–4690.

[24] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. 2017. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4681–4690.

[25] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. 2017. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4681–4690.

[26] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. 2017. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4681–4690.